

Patent Application of

Jeff S. Eder

for

A PERFORMANCE MANAGEMENT PLATFORM

DIVISIONAL AND

CROSS REFERENCE TO RELATED PATENT AND APPLICATION

This application is a divisional of application number 09/994,720 filed November 28, 2001 which is incorporated herein by reference. The subject matter of this application is also related to U.S. Patent 5,615,109 for "Method of and System for Generating Feasible, Profit Maximizing Requisition Sets", by Jeff S. Eder, and U.S. Patent Application 09/940,450 filed August 29, 2001 by Jeff S. Eder the disclosures of which are also incorporated herein by reference.

BACKGROUND OF THE INVENTION

This invention relates to a method of and system for creating a performance management platform for an organization and using said platform to support the analysis, management, optimization, reporting and simulation of the organization or any part of the organization.

SUMMARY OF THE INVENTION

It is a general object of the present invention to provide a novel and useful system for creating a performance management platform and using the platform to optimize the financial performance of an organization.

An object to which the present invention is applied is fully quantifying and then optimizing the assets, derivatives, investments and options associated with operating a commercial organization. Quantification and optimization are enabled by:

- 1) Systematically analyzing up to five segments of value – current operation, real options/contingent liabilities, derivatives, excess financial assets and market sentiment for the organization;
- 2) Systematically analyzing and valuing all the elements of value, tangible and intangible, that affect the segments of value for the organization;
- 3) Optionally analyzing and valuing all the external factors that affect the segments of value for the organization;
- 4) Integrating information from asset management systems (i.e. Customer Relationship Management, Brand Management, etc.), and business intelligence systems for the organization; and
- 5) Integrating the above to construct the performance management platform .

While one embodiment of the novel system for creating a performance management platform analyzes all five segments of value, the system can operate when one or more of the segments of value are not present. For example, the organization may be a private

company that does not have a market value in which case there will be no market sentiment to evaluate. Another common situation would be a corporation that has no derivatives and/or excess financial assets.

As detailed later, the segments of value that are present in the organization are defined in the system settings table (140). Virtually all public companies will have at least three segments of value, current operation, real options and market sentiment. However, it is worth noting only one segment of value is required for operation of the system. The system of the present invention has the added benefit of eliminating a great deal of time-consuming and expensive effort by automating the extraction of data from the databases, tables, and files of existing computer-based corporate finance, operations, human resource, supply chain, web-site and asset management system databases as required to operate the system. In accordance with the invention, the automated extraction, aggregation and analysis of data from a variety of existing computer-based systems significantly increases the scale and scope of the analysis that can be completed. The system of the present invention further enhances the efficiency and effectiveness of the analysis by automating the retrieval, storage and analysis of information useful for valuing elements of value and segments of value from external databases, external publications and the Internet.

Uncertainty over which method is being used for completing the valuation and the resulting inability to compare different valuations is eliminated by the present invention by consistently utilizing the same set of valuation methodologies for valuing the different segments of value as shown in Table 2.

Tabl 2

Segment of organization value	Valuation methodology
• Current-operation value (COPTOT) – value of operation that is developing, making, supplying and selling products and/or services	Income valuation
• Excess net financial assets (aka Excess financial assets)	Total Net Financial Assets valued using GAAP – (amount required to support current operation)
• Real Options & Contingent Liabilities (aka Real options)	Real option algorithms
• Derivatives – includes all hedges, swaps, swaptions, options and warrants	Risk Neutral Valuation
• Market Sentiment	Market Value* – (COPTOT + \sum Real Option Values + \sum Derivatives + \sum Excess Financial Assets)

* The user also has the option of specifying the total value

The market value of the organization is calculated by combining the market value of all debt and equity as shown in Table 3. Element and external factor values are calculated based on the sum of their relative contributions to each segment of value for the organization.

Table 3

Organization Market Value =
\sum Market value of equity for all organizations
-
\sum Market value of debt for all organizations

The utility of the valuations produced by the system of the present invention are further enhanced by explicitly calculating the expected longevity of the different elements of value as required to improve the accuracy and usefulness of the valuations.

As shown in Table 2, real options and contingent liabilities are valued using real option algorithms. Because real option algorithms explicitly recognize whether or not an investment is reversible and/or if it can be delayed, the values calculated using these algorithms are more realistic than valuations created using more traditional approaches

like Net Present Value. The use of real option analysis for valuing growth opportunities and contingent liabilities (hereinafter, real options) gives the present invention a distinct advantage over traditional approaches to organization financial management.

The innovative system has the added benefit of providing a large amount of detailed information to the organization users concerning both tangible and intangible elements of value . Because intangible elements are by definition not tangible, they can not be measured directly. They must instead be measured by the impact they have on their surrounding environment. There are analogies in the physical world. For example, electricity is an “intangible” that is measured by the impact it has on the surrounding environment. Specifically, the strength of the magnetic field generated by the flow of electricity through a conductor turns a motor and the motion of this motor is used to determine the amount of electricity that is being consumed.

The system of the present invention measures intangible elements of value by identifying the attributes that, like the magnetic field, reflect the strength of the element in driving segments of value (current operation, excess financial assets, real options, derivatives, market sentiment) and/or components of value within the current operation (revenue, expense and change in capital) and are relatively easy to measure. Once the attributes related to the strength of each element are identified, they can be summarized into a single expression (a composite variable or vector) if the attributes don't interact with attributes from other elements. If attributes from one element drive those from another, then the elements can be combined for analysis and/or the impact of the individual attributes can be summed together to calculate a value for the element. In one embodiment, vectors are used to summarize the impact of the element attributes. The vectors for all elements are then evaluated to determine their relative contribution to driving each of the components of value and/or each of the segments of value. The system of the present invention calculates the product of the relative contribution and the forecast longevity of each element to determine the relative contribution to each of the components of value to an overall value. The contribution of each element to each component of value are then added together to determine the value of the current operation contribution of each element (see Table 5). The contribution of each element to the organization is then determined by summing the element contribution to each segment of value.

In addition to supporting the identification and display of the efficient frontier, the system of the present invention will also facilitate: analysis of potential mergers and acquisitions, evaluation of asset purchases/disposals, rating the ability of the organization to re-pay debt and monitoring the performance of outside vendors who have been hired boost the value of one or more elements of value (i.e. advertising to increase brand value).

To facilitate its use as a tool for financial management, the system of the present invention produces reports in formats that are similar to the reports provided by traditional accounting systems. Incorporating information regarding all the elements of value is just one of the ways the system of the present invention overcomes the limitations of existing systems.

By providing an real-time financial insight to personnel in the organization, the system of the present invention enables the continuous optimization of management decision making across the entire organization.

BRIEF DESCRIPTION OF DRAWINGS

These and other objects, features and advantages of the present invention will be more readily apparent from the following description of one embodiment of the invention in which:

FIG. 1 is a block diagram showing the major processing steps of the present invention;

FIG. 2 is a diagram showing the files or tables in the application database (50) of the present invention that are utilized for data storage and retrieval during the processing in the innovative system for enterprise analysis and optimization;

FIG. 3 is a block diagram of an implementation of the present invention;

FIG. 4 is a diagram showing the data windows that are used for receiving information from and transmitting information to the user (20) during system processing;

FIG. 5A, FIG. 5B, FIG. 5C, FIG. 5D, FIG. 5E, FIG. 5F and FIG. 5G are block diagrams showing the sequence of steps in the present invention used for specifying system settings and for initializing and operating the data bots that extract, aggregate, store and manipulate information utilized in system processing ;

FIG. 6A, FIG 6B and FIG. 6C are block diagrams showing the sequence of steps in the present invention used for analyzing the value associated with the organization;

FIG. 7 is a block diagram showing the sequence in steps in the present invention used in defining and displaying the matrix of value;

FIG. 8 is a diagram showing an organization matrix of value;

FIG. 9 is a sample Value Map™ Report from the present invention showing the calculated value for the segments of value, the elements of value and the external factors for the organization on the valuation date;

DETAILED DESCRIPTION OF ONE EMBODIMENT

FIG.1 provides an overview of the processing completed by the innovative system for developing and using performance management platform for an organization. In accordance with the present invention, an automated method of and system (100) for producing the performance management platform for a commercial enterprise and using it to support performance improvement activities is provided.

Processing starts in this system (100) with the specification of system settings and the initialization and activation of software data "bots" (200) that extract, aggregate, manipulate and store the data and user (20) input required for completing system processing. This information is extracted via a network (45) from: a basic financial system database (5), an operation management system database (10), a web site transaction log database (12), a human resource information system database (15), an external database (25), an advanced financial system database (30), a asset management system database (35), a supply chain system database (37) and the Internet (40) for the organization. These information extractions and aggregations may be influenced by a user (20) through interaction with a user-interface portion of the application software (700) that mediates the display, transmission and receipt of all information to and from browser software (800) such as the Microsoft Internet Explorer in an access device (90) such as a phone, pda or personal computer that the user (20) interacts with. While only one database of each type (5, 10, 12, 15, 25, 30, 35 and 37) is shown in FIG. 1, it is to be understood that the system (100) can extract data from more than one database of each type via the network (45) for the organization. While the data from multiple asset management systems can be utilized in the analysis of each element of value completed by the system of the present invention, one embodiment of the present invention contains one asset management system for each element of value being analyzed for the organization. Asset management systems can include: customer relationship management systems, partner relationship management systems, channel management systems, knowledge management systems, network analysis systems, visitor relationship management systems, intellectual property management systems, alliance management systems, process management systems, brand management systems, workforce management systems, human resource management systems, email management systems, IT management systems and/or quality management systems.

Automating the extraction and analysis of data from each asset management system ensures that every asset – tangible or intangible - is considered within the overall financial framework for the organization. It should also be understood that it is possible to complete a bulk extraction of data from each database (5, 10, 12, 15, 17, 25, 30, 35 and 37) and the Internet 40 via the network (45) using peer to peer networking and data extraction applications such as Data Transformation Services from Microsoft or the Power Center from Informatica before initializing the data bots. The data extracted in bulk could be stored in a single datamart, a data warehouse or a storage area network where the data bots could operate on the aggregated data.

All extracted information is stored in a file or table (hereinafter, table) within an application database (50) as shown in FIG. 2. The application database (50) contains tables for storing user input, extracted information and system calculations including a system settings table (140), a metadata mapping table (141), a conversion rules table (142), a basic financial system table (143), an operation system table (144), a human resource system table (145), an external database table (146), an advanced finance system table (147), a asset system table (148), a bot date table (149), a keyword table (150), a classified text table (151), a geospatial measures table (152), a composite variables table (153), an industry ranking table (154), an element definition table (155), a segment definition table (156), a cluster ID table (157), an element variables table (158), a vector table (159), a bot table (160), a cash flow table (161), a real option value table (162), a vector table (163), a report table (164), a frame definition table (165), a market sentiment table (166), a value driver change table (167), a simulation table (168), an external factor definition table (169), a statistics table (170), a scenarios table (171), a web log data table (172), a semantic map table (173), a supply chain system table (174), an optimal mix table (175), a factor variables table (176), an xml summary table (177), an analysis definition table (178) and a financial forecasts table. The application database (50) can optionally exist as a datamart, data warehouse or storage area network. The system of the present invention has the ability to accept and store supplemental or primary data directly from user input, a data warehouse or other electronic files in addition to receiving data from the databases described previously. The system of the present invention also has the ability to complete the necessary calculations without receiving data from one or more of the specified databases. However, in one embodiment all required information is obtained from the specified data sources (5, 10, 12, 15, 25, 30, 35, 37 and 40) for the organization.

As shown in FIG. 3, one embodiment of the present invention is a computer system (100) illustratively comprised of a user-interface personal computer (110) connected to an application-server personal computer (120) via a network (45). The application server personal computer (120) is in turn connected via the network (45) to a database-server personal computer (130). The user interface personal computer (110) is also connected via the network (45) to an Internet browser appliance (90) that contains browser software (800) such as Microsoft Internet Explorer or Netscape Navigator.

The database-server personal computer (130) has a read/write random access memory (131), a hard drive (132) for storage of the application database (50), a keyboard (133), a communications bus (134), a display (135), a mouse (136), a CPU (137) and a printer (138).

The application-server personal computer (120) has a read/write random access memory (121), a hard drive (122) for storage of the non-user-interface portion of the application software (200, 300 and 400) of the present invention, a keyboard (123), a communications bus (124), a display (125), a mouse (126), a CPU (127) and a printer (128). While only one client personal computer is shown in FIG. 3, it is to be understood that the application-server personal computer (120) can be networked to fifty or more client, user-interface personal computers (110) via the network (45). The application-server personal computer (120) can also be networked to fifty or more server, personal computers (130) via the network (45). It is to be understood that the diagram of FIG. 3 is merely illustrative of one embodiment of the present invention.

The user-interface personal computer (110) has a read/write random access memory (111), a hard drive (112) for storage of a client data-base (49) and the user-interface portion of the application software (700), a keyboard (113), a communications bus (114), a display (115), a mouse (116), a CPU (117) and a printer (118).

The application software (200, 300 and 400) controls the performance of the central processing unit (127) as it completes the calculations required to support the production of the performance management platform for a commercial enterprise. In the embodiment illustrated herein, the application software program (200, 300, and 400 is written in a combination of C++, Java and Visual Basic®. The application software (200, 300 and 400) can use Structured Query Language (SQL) for extracting data from the databases and the Internet (5, 10, 12, 15, 25, 30, 35, 37 and 40). The user (20) can

optionally interact with the user-interface portion of the application software (700) using the browser software (800) in the browser appliance (90) to provide information to the application software (200, 300, and 400) for use in determining which data will be extracted and transferred to the application database (50) by the data bots.

User input is initially saved to the client database (49) before being transmitted to the communication bus (124) and on to the hard drive (122) of the application-server computer via the network (45). Following the program instructions of the application software, the central processing unit (127) accesses the extracted data and user input by retrieving it from the hard drive (122) using the random access memory (121) as computation workspace in a manner that is well known.

The computers (110, 120, 130 and 139) shown in FIG. 3 illustratively are IBM PCs or clones or any of the more powerful computers or workstations that are widely available. Typical memory configurations for client personal computers (110) used with the present invention should include at least 512 megabytes of semiconductor random access memory (111) and at least a 100 gigabyte hard drive (112). Typical memory configurations for the application-server personal computer (120) used with the present invention should include at least 2056 megabytes of semiconductor random access memory (121) and at least a 250 gigabyte hard drive (122). Typical memory configurations for the database-server personal computer (130) used with the present invention should include at least 4112 megabytes of semiconductor random access memory (131) and at least a 500 gigabyte hard drive (132).

Using the system described above the performance management platform for an organization is produced after the elements of value and external factors are analyzed by segment of value for the organization using the approach outlined in Table 2.

As shown in Table 2, the value of the current-operation for the organization will be calculated using an income valuation. An integral part of most income valuation models is the calculation of the present value of the expected cash flows, income or profits associated with the current-operation. The present value of a stream of cash flows is calculated by discounting the cash flows at a rate that reflects the risk associated with realizing the cash flow. For example, the present value (PV) of a cash flow of ten dollars (\$10) per year for five (5) years would vary depending on the rate used for discounting future cash flows as shown below.

Discount rate = 25%												
PV	=	$\frac{10}{1.25}$	+	$\frac{10}{(1.25)^2}$	+	$\frac{10}{(1.25)^3}$	+	$\frac{10}{(1.25)^4}$	+	$\frac{10}{(1.25)^5}$	=	26.89

Discount rate = 35%												
PV	=	$\frac{10}{1.35}$	+	$\frac{10}{(1.35)^2}$	+	$\frac{10}{(1.35)^3}$	+	$\frac{10}{(1.35)^4}$	+	$\frac{10}{(1.35)^5}$	=	22.20

One of the first steps in evaluating the elements of current-operation value is extracting the data required to complete calculations in accordance with the formula that defines the value of the current-operation as shown in Table 4.

Table 4

Value of current-operation =
(R) Value of forecast revenue from current-operation (positive)
+
(E) Value of forecast expense for current-operation (negative)
+
(C)* Value of current operation capital change forecast

*Note: (C) can have a positive or negative value

The three components of current-operation value will be referred to as the revenue value (R), the expense value (E) and the capital value (C). Examination of the equation in Table 4 shows that there are four ways to increase the value of the current-operation - increase the revenue, decrease the expense, decrease the capital requirements or decrease the interest rate used for discounting future cash flows. As a simplification, the value of the current operation could be calculated from the cash flow which is revenue (a positive number) plus expense (a negative number) and the change in capital (a positive or negative number). A slight adjustment to this basic equation would be required to remove the non-cash depreciation and amortization. The detailed analysis by component of value is utilized in one embodiment.

In one embodiment, the revenue, expense and capital requirement forecasts for the current operation, the real options and the contingent liabilities are obtained from an

advanced financial planning system database (30) derived from an advanced financial planning system similar to the one disclosed in U.S. Patent 5,615,109. The extracted revenue, expense and capital requirement forecasts are used to calculate a cash flow for each period covered by the forecast for the organization by subtracting the expense and change in capital for each period from the revenue for each period. A steady state forecast for future periods is calculated after determining the steady state growth rate that best fits the calculated cash flow for the forecast time period. The steady state growth rate is used to calculate an extended cash flow forecast. The extended cash flow forecast is used to determine the Competitive Advantage Period (CAP) implicit in the organization market value.

While it is possible to use analysis bots to sub-divide each of the components of current operation value into a number of sub-components for analysis, one embodiment has a pre-determined number of sub-components for each component of value for the organization. The revenue value is not subdivided. In one embodiment, the expense value is subdivided into five sub-components: the cost of raw materials, the cost of manufacture or delivery of service, the cost of selling, the cost of support and the cost of administration. The capital value is subdivided into six sub-components: cash, non-cash financial assets, production equipment, other assets (non financial, non production assets), financial liabilities and equity. The components and sub-components of current-operation value will be used in valuing the current operation portion of the elements and sub-elements of value for the organization.

For the calculations completed by the present invention, a transaction will be defined as any event that is logged or recorded. Transaction data is any data related to a transaction. Descriptive data is any data related to any item, segment of value, element of value, component of value or external factor that is logged or recorded. Descriptive data includes forecast data and other data calculated by the system of the present invention. An element of value will be defined as "an entity or group that as a result of past transactions, forecasts or other data has provided and/or is expected to provide economic benefit to the organization." An item will be defined as a single member of the group that defines an element of value. For example, an individual salesman would be an "item" in the "element of value" sales staff. It is possible to have only one item in an element of value. The transaction data and descriptive data associated with an item or related group of items will be referred to as "item variables". Data derived from

transaction data and/or descriptive data are referred to as an item performance indicators. Composite variables for an element are mathematical or logical combinations of item variables and/or item performance indicators. The item variables, item performance indicators and composite variables for a specific element or sub-element of value can be referred to as element variables or element data. External factors are numerical indicators of: conditions or prices external to the organization and conditions or performance of the organization compared to external expectations of conditions or performance. The transaction data and descriptive data associated with external factors will be referred to as "factor variables". Data derived from factor transaction data and/or descriptive data are referred to as factor performance indicators. Composite factors for a factor are mathematical or logical combinations of factor variables and/or factor performance indicators. The factor variables, factor performance indicators and composite factors for external factors can be referred to as factor data.

Analysis bots are used to determine element of value lives and the percentage of: the revenue value, the expense value, and the capital value that are attributable to each element of value . The resulting values are then added together to determine the valuation for different elements as shown by the example in Table 5.

Table 5

Gross Value	Percentage	Element Life/CAP*	Net Value
Revenue value = \$120M	20%	80%	Value = \$19.2 M
Expense value = (\$80M)	10%	80%	Value = (\$6.4) M
Capital value = (\$5M)	5%	80%	Value = (\$0.2) M
Total value = \$35M			
Net value for this element:			Value = \$12.6 M

*CAP = Competitive Advantage Period

The development of the value management matrix for the organization is completed in three distinct stages. As shown in FIG. 5A, FIG. 5B, FIG. 5C, FIG. 5D, FIG. 5E, FIG. 5F and FIG. 5G the first stage of processing (block 200 from FIG. 1) programs bots to continually extract, aggregate, manipulate and store the data from user input, databases and the Internet (5, 10, 12, 15, , 25, 30, 35, 37 and 40) as required for the analysis of business performance . Bots are independent components of the application that have specific tasks to perform. As shown in FIG. 6A, FIG. 6B and FIG. 6C the second stage of

processing (block 300 from FIG. 1) continually values the segments of value and generates a matrix quantifying the impact of elements of value and external factors on the segments of value for each enterprise in the organization (see FIG. 8) by creating and activating analysis bots to:

1. Identify the factor variables, factor performance indicators and composite variables that characterize each external factors impact on: the current operation, derivative and excess financial asset segments of value ,
2. Identify the item variables, item performance indicators and composite variables for each element and sub-element of value that characterize the elements performance in driving: the current operation, derivative and excess financial asset segments of value ,
3. Create vectors that summarize the item variables, item performance indicators and composite variables that define the impact of each element of value and sub-element of value,
4. Create vectors that summarize the factor variables, factor performance indicators and composite variables that define the impact of each external factor,
5. Determine the expected life of each element of value and sub-element of value;
6. Determine the value of the current operation, excess financial assets and derivatives;
7. Determine the appropriate discount rate on the basis of relative causal element strength, value the enterprise real options and contingent liabilities and determine the contribution of each element of value to real option valuation;
8. Determine the best indicator for market price movement, calculate market sentiment and analyze the causes of market sentiment; and
9. Combine the results of the prior stages of processing to determine the value of each external factor, element of value and sub-element of value by segment for each enterprise.

The third stage of processing (block 400 from FIG. 1) displays the matrix of organization value and uses the platform to analyze the impact of changes in structure and/or operation on the financial performance of the organization.

SYSTEM SETTINGS AND DATA BOTS

The flow diagrams in FIG. 5A, FIG. 5B, FIG. 5C, FIG. 5D, FIG. 5E, FIG. 5F and FIG. 5G detail the processing that is completed by the portion of the application software (200) that extracts, aggregates, transforms and stores the information required for system operation from the: basic financial system database (5), operation management system database (10), the web site transaction log database (12), human resource information system database (15), external database (25), advanced financial system database (30), asset management system database (35), the supply chain system database (37), the Internet (40) and the user (20) . A brief overview of the different databases will be presented before reviewing each step of processing completed by this portion (200) of the application software.

Corporate financial software systems are generally divided into two categories, basic and advanced. Advanced financial systems utilize information from the basic financial systems to perform financial analysis, financial planning and financial reporting functions. Virtually every commercial enterprise uses some type of basic financial system, as they are required to use these systems to maintain books and records for income tax purposes. An increasingly large percentage of these basic financial systems are resident in microcomputer and workstation systems. Basic financial systems include general-ledger accounting systems with associated accounts receivable, accounts payable, capital asset, inventory, invoicing, payroll and purchasing subsystems. These systems incorporate worksheets, files, tables and databases. These databases, tables and files contain information about the organization operations and its related accounting transactions. As will be detailed below, these databases, tables and files are accessed by the application software of the present invention as required to extract the information required for completing a business valuation. The system is also capable of extracting the required information from a data warehouse (or datamart) when the required information has been pre-loaded into the warehouse.

General ledger accounting systems generally store only valid accounting transactions. As is well known, valid accounting transactions consist of a debit component and a credit component where the absolute value of the debit component is equal to the absolute value of the credit component. The debits and the credits are posted to the separate accounts maintained within the accounting system. Every basic accounting system has

several different types of accounts. The effect that the posted debits and credits have on the different accounts depends on the account type as shown in Table 6.

Table 6

Account Type:	Debit Impact:	Credit Impact:
Asset	Increase	Decrease
Revenue	Decrease	Increase
Expense	Increase	Decrease
Liability	Decrease	Increase
Equity	Decrease	Increase

General ledger accounting systems also require that the asset account balances equal the sum of the liability account balances and equity account balances at all times.

The general ledger system generally maintains summary, dollar only transaction histories and balances for all accounts while the associated subsystems, accounts payable, accounts receivable, inventory, invoicing, payroll and purchasing, maintain more detailed historical transaction data and balances for their respective accounts. It is common practice for each subsystem to maintain the detailed information shown in Table 7 for each transaction.

Table 7

Subsystem	Detailed Information
Accounts Payable	Vendor, Item(s), Transaction Date, Amount Owed, Due Date, Account Number
Accounts Receivable	Customer, Transaction Date, Product Sold, Quantity, Price, Amount Due, Terms, Due Date, Account Number
Capital Assets	Asset ID, Asset Type, Date of Purchase, Purchase Price, Useful Life, Depreciation Schedule, Salvage Value
Inventory	Item Number, Transaction Date, Transaction Type, Transaction Qty, Location, Account Number
Invoicing	Customer Name, Transaction Date, Product(s) Sold, Amount Due, Due Date, Account Number
Payroll	Employee Name, Employee Title, Pay Frequency, Pay Rate, Account Number
Purchasing	Vendor, Item(s), Purchase Quantity, Purchase Price(s), Due Date, Account Number

As is well known, the output from a general ledger system includes income statements, balance sheets and cash flow statements in well defined formats which assist management in measuring the financial performance of the firm during the prior periods when data input and system processing have been completed.

While basic financial systems are similar between firms, operation management systems vary widely depending on the type of company they are supporting. These systems typically have the ability to not only track historical transactions but to forecast future performance. For manufacturing firms, operation management systems such as Enterprise Resource Planning Systems (ERP), Material Requirement Planning Systems (MRP), Purchasing Systems, Scheduling Systems and Quality Control Systems are used to monitor, coordinate, track and plan the transformation of materials and labor into products. Systems similar to the one described above may also be useful for distributors to use in monitoring the flow of products from a manufacturer.

Operation Management Systems in manufacturing firms may also monitor information relating to the production rates and the performance of individual production workers, production lines, work centers, production teams and pieces of production equipment including the information shown in Table 8.

Table 8

Operation Management System – Production Information	
1. ID number (employee id/machine id)	10. Cumulative training time
2. Actual hours – last batch	11. Job(s) certifications
3. Standard hours – last batch	12. Actual scrap - last batch
4. Actual hours – year to date	13. Scrap allowance - last batch
5. Actual/Standard hours - year to date %	14. Actual scrap/allowance - year to date
6. Actual setup time - last batch	15. Rework time/unit last batch
7. Standard setup time - last batch	16. Rework time/unit year to date
8. Actual setup hours - year to date	17. QC rejection rate - batch
9. Actual/Standard setup hrs - yr to date %	18. QC rejection rate - year to date

Operation management systems are also useful for tracking requests for service to repair equipment in the field or in a centralized repair facility. Such systems generally store information similar to that shown below in Table 9.

Table 9

Operation Management System - Service Call Information	
1. Customer name	11. Promised type of response
2. Customer number	12. Time person dispatched to call
3. Contract number	13. Name of person handling call
4. Service call number	14. Time of arrival on site
5. Time call received	15. Time of repair completion
6. Product(s) being fixed	16. Actual response type
7. Serial number of equipment	17. Part(s) replaced
8. Name of person placing call	18. Part(s) repaired
9. Name of person accepting call	19. 2nd call required
10. Promised response time	20. 2nd call number

Web site transaction log databases keep a detailed record of every visit to a web site, they can be used to trace the path of each visitor to the web site and upon further analysis can be used to identify patterns that are most likely to result in purchases and those that are most likely to result in abandonment. This information can also be used to identify which promotion would generate the most value for the organization using the system. Web site transaction logs generally contain the information shown in Table 10.

Table 10

Web Site Transaction Log Database	
1. Customer's URL	6. Referring site
2. Date and time of visit	7. URL of site visited next
3. Pages visited	8. Downloaded file volume and type
4. Length of page visit (time)	9. Cookies
5. Type of browser used	10. Transactions

Computer based human resource systems may some times be packaged or bundled within enterprise resource planning systems such as those available from SAP, Oracle and Peoplesoft. Human resource systems are increasingly used for storing and maintaining corporate records concerning active employees in sales, operations and the other functional specialties that exist within a modern corporation. Storing records in a centralized system facilitates timely, accurate reporting of overall manpower statistics to the corporate management groups and the various government agencies that require

periodic updates. In some cases, human resource systems include the organization payroll system as a subsystem. In one embodiment of the present invention, the payroll system is part of the basic financial system. These systems can also be used for detailed planning regarding future manpower requirements. Human resource systems typically incorporate worksheets, files, tables and databases that contain information about the current and future employees. As will be detailed below, these databases, tables and files are accessed by the application software of the present invention as required to extract the information required for completing a business valuation. It is common practice for human resource systems to store the information shown in Table 11 for each employee.

Table 11

Human Resource System Information	
1. Employee name	10. Employee start date - enterprise
2. Job title	11. Employee start date - department
3. Job code	12. Employee start date - current job
4. Rating	13. Training courses completed
5. Division	14. Cumulative training expenditures
6. Department	15. Salary history
7. Employee No./(Social Security Number)	16. Current salary
8. Year to date - hours paid	17. Educational background
9. Year to date - hours worked	18. Current supervisor

External databases (25) can be used for obtaining information that enables the definition and evaluation of a variety of things including elements of value, external factors, industry real options and market sentiment. In some cases, information from these databases can be used to supplement information obtained from the other databases and the Internet (5, 10, 12, 15, 30, 35, 37 and 40). In the system of the present invention, the information extracted from external databases (25) can be in the forms listed in Table 12.

Tabl 12

Types of information
1) numeric information such as that found in the SEC Edgar database and the databases of financial infomediaries such as FirstCall, IBES and Compustat,
2) text information such as that found in the Lexis Nexis database and databases containing past issues from specific publications,
3) geospatial data; and
4) multimedia information such as video and audio clips.

The system of the present invention uses different "bot" types to process each distinct data type from external databases (25). The same "bot types" are also used for extracting each of the different types of data from the Internet (40). The system of the present invention must have access to at least one external database (25) that provides information regarding the equity prices for the organization and the equity prices and financial performance of the competitors for the organization.

Advanced financial systems may also use information from external databases (25) and the Internet (40) in completing their processing. Advanced financial systems include financial planning systems and activity based costing systems. Activity based costing systems may be used to supplement or displace the operation of the expense component analysis segment of the present invention. Financial planning systems generally use the same format used by basic financial systems in forecasting income statements, balance sheets and cash flow statements for future periods. Management uses the output from financial planning systems to highlight future financial difficulties with a lead time sufficient to permit effective corrective action and to identify problems in organization operations that may be reducing the profitability of the business below desired levels. These systems are most often developed by individuals within companies using two and three-dimensional spreadsheets such as Lotus 1-2-3®, Microsoft Excel® and Quattro Pro®. In some cases, financial planning systems are built within an executive information system (EIS) or decision support system (DSS). For one embodiment of the present invention, the advanced finance system database is similar to the financial planning system database detailed in U.S. Patent 5,165,109 for "Method of and System for Generating

Feasible, Profit Maximizing Requisition Sets”, by Jeff S. Eder, the disclosure of which is incorporated herein by reference.

While advanced financial planning systems have been around for some time, asset management systems are a relatively recent development. Their appearance is further proof of the increasing importance of “soft” assets. Asset management systems include: customer relationship management systems, partner relationship management systems, channel management systems, knowledge management systems, visitor relationship management systems, intellectual property management systems, investor management systems, vendor management systems, alliance management systems, process management systems, brand management systems, workforce management systems, human resource management systems, email management systems, IT management systems and/or quality management systems. As defined for this application, asset management system data includes all unclassified text and multi-media data within an enterprise or organization. Asset management systems are similar to operation management systems in that they generally have the ability to forecast future events as well as track historical occurrences. Many have also added analytical capabilities that allow them to identify trends and patterns in the data associated with the asset they are managing. Customer relationship management systems are the most well established asset management systems at this point and will be the focus of the discussion regarding asset management system data. In firms that sell customized products, the customer relationship management system is generally integrated with an estimating system that tracks the flow of estimates into quotations, orders and eventually bills of lading and invoices. In other firms that sell more standardized products, customer relationship management systems generally are used to track the sales process from lead generation to lead qualification to sales call to proposal to acceptance (or rejection) and delivery. All customer relationship management systems would be expected to track all of the customer’s interactions with the organization after the first sale and store information similar to that shown below in Table 13.

Tabl 13

Custom r R lationship Manag m nt Syst m – Inf rmation	
1. Customer/Potential customer name	9. Sales call history
2. Customer number	10. Sales contact history
3. Address	11. Sales history: product/qty/price
4. Phone number	12. Quotations: product/qty/price
5. Source of lead	13. Custom product percentage
6. Date of first purchase	14. Payment history
7. Date of last purchase	15. Current A/R balance
8. Last sales call/contact	16. Average days to pay

Supply chain systems could be considered as asset management systems as they are used to manage a critical asset – supplier relationships. However, because of their importance and visibility they are listed separately. Supply chain management system databases (37) contain information that may have been in operation management system databases (10) in the past. These systems provide enhanced visibility into the availability of goods and promote improved coordination between customers and their suppliers. All supply chain management systems would be expected to track all of the items ordered by the organization after the first purchase and store information similar to that shown below in Table 14.

Table 14

Supply Chain Management System Information	
1. Stock Keeping Unit (SKU)	7. Quantity available today
2. Vendor	8. Quantity available next 7 days
3. Total Quantity on Order	9. Quantity available next 30 days
4. Total Quantity in Transit	10. Quantity available next 90 days
5. Total Quantity on Back Order	11. Quoted lead time
6. Total Quantity in Inventory	12. Actual average lead time

System processing of the information from the different databases (5, 10, 12, 15, 17, 25, 30, 35 and 37) and the Internet (40) described above starts in a block 201, FIG. 5A, which immediately passes processing to a software block 202. The software in block 202 prompts the user (20) via the system settings data window (701) to provide system setting

information. The system setting information entered by the user (20) is transmitted via the network (45) back to the application server (120) where it is stored in the system settings table (140) in the application database (50) in a manner that is well known. The specific inputs the user (20) is asked to provide at this point in processing are shown in Table 15.

Table 15

1. New calculation or structure revision?
2. Continuous, If yes, new calculation frequency? (hourly, daily, weekly, monthly or quarterly)
3. Structure of organization (enterprises, segments of value, elements of value etc.)
4. Organization checklist
5. Base account structure
6. Base currency
7. Location of account structure
8. Metadata standard
9. Location of basic financial system database and metadata
10. Location of advanced finance system database and metadata
11. Location of human resource information system database and metadata
12. Location of operation management system database and metadata
13. Location of asset management system databases and metadata
14. Location of external databases and metadata
15. Location of web site transaction log database and metadata
16. Location of supply chain management system database and metadata
17. Location of database and metadata for equity information
18. Location of database and metadata for debt information
19. Location of database and metadata for tax rate information
20. Location of database and metadata for currency conversion rate information
21. Geospatial data? If yes, identity of geocoding service.
22. The maximum number of generations to be processed without improving fitness
23. Default clustering algorithm (selected from list) and maximum cluster number
24. Minimum amount of cash and marketable securities required for operations
25. Total cost of capital (weighted average cost of equity, debt and risk capital)
26. Number of months a product is considered new after it is first produced
27. Organization industry classification (SIC Code)
28. Primary competitors by industry classification
29. Management report types (text, graphic, both)
30. Default Missing Data Procedure
31. Maximum time to wait for user input

- 32. Maximum discount rate for new projects (real option valuation)
- 33. Detailed valuation using components of current operation value? (yes or no)
- 34. Use of industry real options? (yes or no)
- 35. Maximum number of sub-elements
- 36. Minimum working capital level (optional)
- 37. Semantic mapping? (yes or no)

The organization checklist data are used by a “rules” engine (such as the one available from Neuron Data) in block 202 to influence the number and type of items with pre-defined metadata mapping for each category of value. For example, if the checklist data indicates that the organization is focused on branded, consumer markets, then additional brand related factors would be pre-defined for mapping. The application of these system settings will be further explained as part of the detailed explanation of the system operation.

The software in block 202 uses the current system date to determine the time periods (months) that require data to complete the calculations. After the date range is calculated it is stored in the system settings table (140). In one embodiment the analysis of organization value by the system obtains and utilizes data from every source for the four year period before and the three year forecast period after the specified valuation date and/or the date of system calculation. The system is capable of processing data up to 10 years before the date of system calculation (hereinafter referred to as the system date) and 5 years after the date of system calculation and can function when less data is available. The user (20) also has the option of specifying the data periods that will be used for completing system calculations.

After the storage of system setting data is complete, processing advances to a software block 203. The software in block 203 prompts the user (20) via the metadata and conversion rules window (702) to map metadata using the standard specified by the user (20) from the basic financial system database (5), the operation management system database (10), the web site transaction log database (12), the human resource information system database (15), the external database (25), the advanced financial system database (30), the asset management system database (35) and the supply chain system database (37) to the organization hierarchy stored in the system settings table (140) and to the pre-specified fields in the metadata mapping table (141). Pre-specified fields in the metadata mapping table include the revenue, expense and capital

components and sub-components of current operation value for the organization and pre-specified fields for expected value drivers by element of value and external factor. Because the bulk of the information being extracted is financial information, the metadata mapping often takes the form of specifying the account number ranges that correspond to the different fields in the metadata mapping table (141). Table 16 shows the base account number structure that the account numbers in the other systems must align with. For example, using the structure shown below, the revenue component for the organization could be specified as enterprise 01, any department number, accounts 400 to 499 (the revenue account range) with any sub-account.

Table 16

Account Number	01 -	902 (any) -	477-	86 (any)
Section	Enterprise	Department	Account	Sub-account
Subgroup	Workstation	Marketing	Revenue	Singapore
Position	4	3	2	1

As part of the metadata mapping process, any database fields that are not mapped to pre-specified fields are defined by the user (20) as component of value, elements of value or non-relevant attributes and “mapped” in the metadata mapping table (141) to the corresponding fields in each database in a manner identical to that described above for the pre-specified fields. After all fields have been mapped to the metadata mapping table (141), the software in block 203 prompts the user (20) via the metadata and conversion rules window (702) to provide conversion rules for each metadata field for each data source. Conversion rules will include information regarding currency conversions and conversion for units of measure that may be required to accurately and consistently analyze the data. The inputs from the user (20) regarding conversion rules are stored in the conversion rules table (142) in the application database (50). When conversion rules have been stored for all fields from every data source, then processing advances to a software block 204.

The software in block 204 checks the system settings table (140) in the application database (50) to determine if the current calculation is a new calculation or a structure change. The calculation (or run) may be new because the system is running for first time or it may be because the sytem is running continuously and it is now time for a new

calculation to be completed. If the calculation is not a new calculation or a structure change then processing advances to a software block 212. Alternatively, if the calculation is new or a structure change, then processing advances to a software block 207.

The software in block 207 checks the bot date table (149) and deactivates any basic financial system data bots with creation dates before the current system date and retrieves information from the system settings table (140), metadata mapping table (141) and conversion rules table (142). The software in block 207 then initializes data bots for each field in the metadata mapping table (141) that mapped to the basic financial system database (5) in accordance with the frequency specified by user (20) in the system settings table (140). Bots are independent components of the application that have specific tasks to perform. In the case of data acquisition bots, their tasks are to extract and convert transaction and descriptive data from a specified source and then store it in a specified location. Each data bot initialized by software block 207 will store its data in the basic financial system table (143) and/or the derivatives table (175). Every data acquisition bot contains the information shown in Table 17.

Table 17

1. Unique ID number (based on date, hour, minute, second of creation)
2. The data source location
3. Mapping information
4. Timing of extraction
5. Conversion rules (if any)
6. Storage Location (to allow for tracking of source and destination events)
7. Enterprise
8. Creation date (date, hour, minute, second)

After the software in block 207 initializes all the bots for the basic financial system database, processing advances to a block 208. In block 208, the bots extract and convert transaction and descriptive data from the basic financial system (5) in accordance with their preprogrammed instructions in accordance with the frequency specified by user (20) in the system settings table (140). As each bot extracts and converts data from the basic financial system database (5), processing advances to a software block 209 before the bot completes data storage. The software in block 209 checks the basic financial system

metadata to see if all fields have been extracted. If the software in block 209 finds no unmapped data fields, then the extracted, converted data are stored in the basic financial system table (143) . Alternatively, if there are fields that have not been extracted, then processing advances to a block 211. The software in block 211 prompts the user (20) via the metadata and conversion rules window (702) to provide metadata and conversion rules for each new field. The information regarding the new metadata and conversion rules is stored in the metadata mapping table (141) and conversion rules table (142) while the extracted, converted data are stored in the basic financial system table (143) . It is worth noting at this point that the activation and operation of bots where all the fields have been mapped to the application database (50) continues. Only bots with unmapped fields “wait” for user input before completing data storage. The new metadata and conversion rule information will be used the next time bots are initialized in accordance with the frequency established by the user (20). In either event, system processing passes on to software block 212.

The software in block 212 checks the system settings table (140) in the application database (50) to determine if the current calculation is a new calculation or a structure change. If the calculation is not a new calculation or a structure change then processing advances to a software block 228. Alternatively, if the calculation is new or a structure change, then processing advances to a software block 221.

The software in block 221 checks the bot date table (149) and deactivates any operation management system data bots with creation dates before the current system date and retrieves information from the system settings table (140), metadata mapping table (141) and conversion rules table (142). The software in block 221 then initializes data bots for each field in the metadata mapping table (141) that mapped to the operation management system database (10) in accordance with the frequency specified by user (20) in the system settings table (140). Each data bot initialized by software block 221 will store its data in the operation system table (144) .

After the software in block 221 initializes all the bots for the operation management system database, processing advances to a block 222. In block 222, the bots extract and convert transaction and descriptive data from the operation management system database (10) in accordance with their preprogrammed instructions in accordance with the frequency specified by user (20) in the system settings table (140). As each bot

extracts and converts data from the operation management system database (10), processing advances to a software block 209 before the bot completes data storage. The software in block 209 checks the operation management system metadata to see if all fields have been extracted. If the software in block 209 finds no unmapped data fields, then the extracted, converted data are stored in the operation system table (144) . Alternatively, if there are fields that have not been extracted, then processing advances to a block 211. The software in block 211 prompts the user (20) via the metadata and conversion rules window (702) to provide metadata and conversion rules for each new field. The information regarding the new metadata and conversion rules is stored in the metadata mapping table (141) and conversion rules table (142) while the extracted, converted data are stored in the operation system table (144) . It is worth noting at this point that the activation and operation of bots where all the fields have been mapped to the application database (50) continues. Only bots with unmapped fields “wait” for user input before completing data storage. The new metadata and conversion rule information will be used the next time bots are initialized in accordance with the frequency established by the user (20). In either event, system processing then passes on to a software block 225.

The software in block 225 checks the bot date table (149) and deactivates any web site transaction log data bots with creation dates before the current system date and retrieves information from the system settings table (140), metadata mapping table (141) and conversion rules table (142). The software in block 225 then initializes data bots for each field in the metadata mapping table (141) that mapped to the web site transaction log database (12) in accordance with the frequency specified by user (20) in the system settings table (140). Each data bot initialized by software block 225 will store its data in the web log data table (172) .

After the software in block 225 initializes all the bots for the web site transaction log database, the bots extract and convert transaction and descriptive data in accordance with their preprogrammed instructions in accordance with the frequency specified by user (20) in the system settings table (140). As each bot extracts and converts data from the web site transaction log database (12), processing advances to a software block 209 before the bot completes data storage. The software in block 209 checks the web site transaction log metadata to see if all fields have been extracted. If the software in block 209 finds no unmapped data fields, then the extracted, converted data are stored in the

web log data table (172) . Alternatively, if there are fields that have not been extracted, then processing advances to a block 211. The software in block 211 prompts the user (20) via the metadata and conversion rules window (702) to provide metadata and conversion rules for each new field. The information regarding the new metadata and conversion rules is stored in the metadata mapping table (141) and conversion rules table (142) while the extracted, converted data are stored in the web log data table (172) . It is worth noting at this point that the activation and operation of bots where all the fields have been mapped to the application database (50) continues. Only bots with unmapped fields “wait” for user input before completing data storage. The new metadata and conversion rule information will be used the next time bots are initialized in accordance with the frequency established by the user (20). In either event, system processing then passes on to a software block 226.

The software in block 226 checks the bot date table (149) and deactivates any human resource information system data bots with creation dates before the current system date and retrieves information from the system settings table (140), metadata mapping table (141) and conversion rules table (142). The software in block 226 then initializes data bots for each field in the metadata mapping table (141) that mapped to the human resource information system database (15) in accordance with the frequency specified by user (20) in the system settings table (140). Each data bot initialized by software block 226 will store its data in the human resource system table (145) .

After the software in block 226 initializes all the bots for the human resource information system database, the bots extract and convert transaction and descriptive data in accordance with their preprogrammed instructions in accordance with the frequency specified by user (20) in the system settings table (140) . As each bot extracts and converts data from the human resource information system database (15), processing advances to a software block 209 before the bot completes data storage. The software in block 209 checks the human resource information system metadata to see if all fields have been extracted. If the software in block 209 finds no unmapped data fields, then the extracted, converted data are stored in the human resource system table (145) . Alternatively, if there are fields that haven't been extracted, then processing advances to a block 211. The software in block 211 prompts the user (20) via the metadata and conversion rules window (702) to provide metadata and conversion rules for each new field. The information regarding the new metadata and conversion rules is stored in the

metadata mapping table (141) and conversion rules table (142) while the extracted, converted data are stored in the human resource system table (145) . It is worth noting at this point that the activation and operation of bots where all the fields have been mapped to the application database (50) continues. Only bots with unmapped fields “wait” for user input before completing data storage. The new metadata and conversion rule information will be used the next time bots are initialized in accordance with the frequency established by the user (20). In either event, system processing then passes on to software block 228.

The software in block 228 checks the system settings table (140) in the application database (50) to determine if the current calculation is a new calculation or a structure change. If the calculation is not a new calculation or a structure change then processing advances to a software block 248. Alternatively, if the calculation is new or a structure change, then processing advances to a software block 241.

The software in block 241 checks the bot date table (149) and deactivates any external database data bots with creation dates before the current system date and retrieves information from the system settings table (140), metadata mapping table (141) and conversion rules table (142). The software in block 241 then initializes data bots for each field in the metadata mapping table (141) that mapped to the external database (25) in accordance with the frequency specified by user (20) in the system settings table (140). Each data bot initialized by software block 241 will store its data in the external database table (146) .

After the software in block 241 initializes all the bots for the external database, processing advances to a block 242. In block 242, the bots extract, convert and assign transaction and descriptive data in accordance with their preprogrammed instructions. As each bot extracts, converts and assigns data from the external database (25), processing advances to a software block 209 before the bot completes data storage and assignments. The software in block 209 checks the external database metadata to see if the extracted data are assigned to specified fields. If the software in block 209 finds no unmapped data, then the extracted, converted data are stored in the external database table (146). Alternatively, if there are fields that do not have metadata assignments, then processing advances to a block 211. The software in block 211 prompts the user (20) via the metadata and conversion rules window (702) to provide metadata, conversion

rules and assignments for each new field. The information regarding the new metadata and conversion rules is stored in the metadata mapping table (141) and conversion rules table (142) while the information regarding the new assignments is stored in the external factor definition table (169). While some external factors are pre-defined for analysis, the bulk of the external factors are not pre-assigned and are developed using available data that is assigned to an external factor at the time of extraction. The extracted, converted data with new assignments is then stored in the external database table (146). It is worth noting at this point that the activation and operation of bots where all the fields have been mapped to the application database (50) continues. Only bots with unmapped fields "wait" for user input before completing data storage. The new metadata, conversion rule and classification information will be used the next time bots are initialized in accordance with the frequency established by the user (20). In either event, system processing then passes on to a software block 245.

The software in block 245 checks the bot date table (149) and deactivates any advanced financial system data bots with creation dates before the current system date and retrieves information from the system settings table (140), metadata mapping table (141) and conversion rules table (142). The software in block 245 then initializes data bots for each field in the metadata mapping table (141) that mapped to the advanced financial system database (30) in accordance with the frequency specified by user (20) in the system settings table (140). Each data bot initialized by software block 245 will store its data in the advanced finance system database table (147).

After the software in block 245 initializes all the bots for the advanced finance system database, the bots extract and convert transaction and descriptive data in accordance with their preprogrammed instructions in accordance with the frequency specified by user (20) in the system settings table (140). As each bot extracts and converts data from the advanced financial system database (30), processing advances to a software block 209 before the bot completes data storage. The software in block 209 checks the advanced finance system database metadata to see if all fields have been extracted. If the software in block 209 finds no unmapped data fields, then the extracted, converted data are stored in the advanced finance system database table (147). Alternatively, if there are fields that haven't been extracted, then processing advances to a block 211. The software in block 211 prompts the user (20) via the metadata and conversion rules window (702) to provide metadata and conversion rules for each new field. The information regarding the

new metadata and conversion rules is stored in the metadata mapping table (141) and conversion rules table (142) while the extracted, converted data are stored in the advanced finance system database table (147) . It is worth noting at this point that the activation and operation of bots where all the fields have been mapped to the application database (50) continues. Only bots with unmapped fields “wait” for user input before completing data storage. The new metadata and conversion rule information will be used the next time bots are initialized in accordance with the frequency established by the user (20). In either event, system processing then passes on to software block 246.

The software in block 246 checks the bot date table (149) and deactivates any asset management system data bots with creation dates before the current system date and retrieves information from the system settings table (140), metadata mapping table (141) and conversion rules table (142). The software in block 246 then initializes data bots for each field in the metadata mapping table (141) that mapped to a asset management system database (35) in accordance with the frequency specified by user (20) in the system settings table (140). Extracting data from each asset management system ensures that the management of each soft asset is considered and prioritized within the overall financial models for the organization. Each data bot initialized by software block 246 will store its data in the asset system table (148) .

After the software in block 246 initializes bots for all asset management system databases, the bots extract and convert transaction and descriptive data in accordance with their preprogrammed instructions in accordance with the frequency specified by user (20) in the system settings table (140). As each bot extracts and converts data from the asset management system databases (35), processing advances to a software block 209 before the bot completes data storage. The software in block 209 checks the metadata for the asset management system databases to see if all fields have been extracted. If the software in block 209 finds no unmapped data fields, then the extracted, converted data are stored in the asset system table (148) . Alternatively, if there are fields that haven't been extracted, then processing advances to a block 211. The software in block 211 prompts the user (20) via the metadata and conversion rules window (702) to provide metadata and conversion rules for each new field. The information regarding the new metadata and conversion rules is stored in the metadata mapping table (141) and conversion rules table (142) while the extracted, converted data are stored in the asset system table (148) . It is worth noting at this point that the activation and operation of

bots where all the fields have been mapped to the application database (50) continues. Only bots with unmapped fields "wait" for user input before completing data storage. The new metadata and conversion rule information will be used the next time bots are initialized in accordance with the frequency established by the user (20). In either event, system processing then passes on to software block 248.

The software in block 248 checks the system settings table (140) in the application database (50) to determine if the current calculation is a new calculation or a structure change. If the calculation is not a new calculation or a structure change then processing advances to a software block 254. Alternatively, if the calculation is new or a structure change, then processing advances to a software block 252.

The software in block 252 checks the bot date table (149) and deactivates any supply chain system data bots with creation dates before the current system date and retrieves information from the system settings table (140), metadata mapping table (141) and conversion rules table (142). The software in block 252 then initializes data bots for each field in the metadata mapping table (141) that mapped to a supply chain system database (37) in accordance with the frequency specified by user (20) in the system settings table (140). Each data bot initialized by software block 252 will store its data in the supply chain system table (174).

After the software in block 252 initializes bots for all supply chain system databases, the bots extract and convert transaction and descriptive data in accordance with their preprogrammed instructions in accordance with the frequency specified by user (20) in the system settings table (140). As each bot extracts and converts data from the supply chain system databases (37), processing advances to a software block 209 before the bot completes data storage. The software in block 209 checks the metadata for the supply chain system database (37) to see if all fields have been extracted. If the software in block 209 finds no unmapped data fields, then the extracted, converted data are stored in the supply chain system table (174). Alternatively, if there are fields that have not been extracted, then processing advances to a block 211. The software in block 211 prompts the user (20) via the metadata and conversion rules window (702) to provide metadata and conversion rules for each new field. The information regarding the new metadata and conversion rules is stored in the metadata mapping table (141) and conversion rules table (142) while the extracted, converted data are stored in the supply

chain system table (174) . It is worth noting at this point that the activation and operation of bots where all the fields have been mapped to the application database (50) continues. Only bots with unmapped fields “wait” for user input before completing data storage. The new metadata and conversion rule information will be used the next time bots are initialized in accordance with the frequency established by the user (20). In either event, system processing then passes on to software block 254.

The software in block 254 checks the system settings table (140) in the application database (50) to determine if the current calculation is a new calculation or a structure change. If the calculation is not a new calculation or a structure change then processing advances to a software block 276. Alternatively, if the calculation is new or a structure change, then processing advances to a software block 255.

The software in block 255 prompts the user (20) via the identification and classification rules window (703) to identify keywords such as company names, brands, trademarks, competitors, risks and trends for pre-specified fields in the metadata mapping table (141) . After specifying the keywords, the user (20) is prompted to classify each keyword by element, factor, enterprise or industry (note more than one classification per keyword is possible). The classification information provided by the user (20) is supplemented by a second classification that identifies the semantic map or maps associated with the keyword. The input from the user (20) is stored in the keyword table (150) in the application database before processing advances to a software block 257.

The software in block 257 checks the bot date table (149) and deactivates any Internet text and linkage bots with creation dates before the current system date and retrieves information from the system settings table (140), the metadata mapping table (141) and the keyword table (150). The software in block 257 then initializes Internet text and linkage bots for each field in the metadata mapping table (141) that mapped to a keyword in accordance with the frequency specified by user (20) in the system settings table (140).

Bots are independent components of the application that have specific tasks to perform. In the case of text and linkage bots, their tasks are to locate, count, classify and extract keyword matches and linkages from the Internet and then store their findings as item variables in a specified location. The classification includes both the factor, element, organization or industry that the keyword is associated with and the context of the keyword mention. This dual classification allows the system of the present invention to

identify both the number of times an organization element was mentioned and the context in which the organization element appeared. For example, the system might identify the fact that an organization brand was mentioned 367 times in the most recent month and that 63% of the mentions were associated with a favorable semantic map. Each Internet text and linkage bot initialized by software block 257 will store the extracted data and the location, count and classification data it discovers in the classified text table (151) . Multimedia data can be processed using these same bots if software to translate and parse the multimedia content is included in each bot. Every Internet text and linkage bot contains the information shown in Table 18.

Table 18

1. Unique ID number (based on date, hour, minute, second of creation)
2. Creation date (date, hour, minute, second)
3. Storage location
4. Mapping information
5. Home URL
6. Enterprise
7. Keyword
8. Element, factor, organization or industry
9. Semantic map

After being initialized, the text and linkage bots locate and classify data from the Internet (40) in accordance with their programmed instructions with the frequency specified by user (20) in the system settings table (140). As each text bot locates and classifies data from the Internet (40) processing advances to a software block 258 before the bot completes data storage. The software in block 258 checks to see if all linkages keyword hits have been classified by element, factor or organization. If the software in block 258 does not find any unclassified "hits" or "links", then the address, counts, dates and classified text are stored in the classified text table (151) . Alternatively, if there are hits or links that haven't been classified, then processing advances to a block 259. The software in block 259 prompts the user (20) via the identification and classification rules window (703) to provide classification rules for each new hit or link. The information regarding the new classification rules is stored in the keyword table (150) while the newly classified text and linkages are stored in the classified text table (151) . It is worth noting

at this point that the activation and operation of bots where all fields map to the application database (50) continues. Only bots with unclassified fields will “wait” for user input before completing data storage. The new classification rules will be used the next time bots are initialized in accordance with the frequency established by the user (20). In either event, system processing then passes on to a software block 263.

The software in block 263 checks the bot date table (149) and deactivates any text bots with creation dates before the current system date and retrieves information from the system settings table (140), the metadata mapping table (141) and the keyword table (150). The software in block 263 then initializes text bots for each field in the metadata mapping table (141) that mapped to a keyword in accordance with the frequency specified by user (20) in the system settings table (140). The text bots use the same classification schema used for Internet text found in external and internal databases. Every bot initialized by software block 263 will store the extracted location, count, date and classification of data it discovers as item variables in the classified text table (151). Every text bot contains the information shown in Table 19.

Table 19

1. Unique ID number (based on date, hour, minute, second of creation)
2. Creation date (date, hour, minute, second)
3. Storage location
4. Mapping information
5. Enterprise
6. Data source
7. Keyword
8. Storage location
9. Element, factor, organization or industry
10. Semantic map

After being initialized, the bots locate data from the external database (25) or an asset management system database (35) in accordance with its programmed instructions with the frequency specified by user (20) in the system settings table (140). As each bot locates and extracts text data, processing advances to a software block 258 before the

bot completes data storage. The software in block 258 checks to see if all keyword hits are classified by element, factor, organization, industry and semantic map. If the software in block 258 does not find any unclassified "hits", then the address, count and classified text are stored in the classified text table (151). Alternatively, if there are terms that have not been classified, then processing advances to a block 259. The software in block 259 prompts the user (20) via the identification and classification rules window (703) to provide classification rules for each new term. The information regarding the new classification rules is stored in the keyword table (150) while the newly classified text is stored in the classified text table (151). It is worth noting at this point that the activation and operation of bots with classified data (50) continues. Only bots with unclassified fields "wait" for user input before completing data storage. The new classification rules will be used the next time bots are initialized in accordance with the frequency established by the user (20). In either event, system processing then passes on to software block 264.

The software in block 264 checks the system settings table (140) to see if there is geospatial data in the application database (50) and to determine which on-line geocoding service (Centrus™ from QM Soft or MapMarker™ from MapInfo) is being used. If geospatial data are not being used, then processing advances to a block 269. Alternatively, if the software in block 264 determines that geospatial data are being used, processing advances to a software block 265.

The software in block 265 prompts the user (20) via the geospatial measure definitions window (710) to define the measures that will be used in evaluating the elements of value. After specifying the measures, the user (20) is prompted to select geospatial loci for each measure from the data already stored in the application database (50). The input from the user (20) is stored in the geospatial measures table (152) in the application database before processing advances to a software block 266.

The software in block 266 checks the bot date table (149) and deactivates any geospatial bots with creation dates before the current system date and retrieves information from the system settings table (140), the metadata mapping table (141) and the geospatial measures table (152). The software in block 266 then initializes geospatial bots for each field in the metadata mapping table (141) that mapped to geospatial data in

the application database (50) in accordance with the frequency specified by user (20) in the system settings table (140) before advancing processing to a software block 280.

Bots are independent components of the application that have specific tasks to perform. In the case of geospatial bots, their tasks are to calculate item variables using a specified geocoding service, then store the measures in a specified location. Each geospatial bot initialized by software block 266 will store the item variable measures it calculates in the application database table where the geospatial data was found. For example, calculated item variables related to customer locations would be stored in the asset management system table (148) for customer data. Tables that are likely to include geospatial data include: the basic financial system table (143), the operation system table (144), the human resource system table (145), the external database table (146), the advanced finance system table (147) and the asset system table (148). Every geospatial bot contains the information shown in Table 20.

Table 20

1. Unique ID number (based on date, hour, minute, second of creation)
2. Creation date (date, hour, minute, second)
3. Mapping information
4. Storage location
5. Enterprise
6. Geospatial locus
7. Geospatial measure
8. Geocoding service

After being activated, the geospatial bots locate data and calculate measurements (which are descriptive item variables) in accordance with their programmed instructions with the frequency specified by the user (20) in the system settings table (140). As each geospatial bot retrieves data and calculates the geospatial measures that have been specified, processing advances to a block 267 before the bot completes data storage. The software in block 267 checks to see if all geospatial data located by the bot have been measured. If the software in block 267 does not find any uncalculated measurement data, then the measurements are stored in the application database (50).

Alternatively, if there are data elements where measures have not been calculated, then processing advances to a block 268. The software in block 268 prompts the user (20) via the geospatial measure definition window (710) to provide measurement rules for each new term. The information regarding the new measurement rules is stored in the geospatial measures table (152) while the newly calculated measurements are stored in the appropriate table in the application database (50). It is worth noting at this point that the activation and operation of bots that do not have unmeasured fields continues. Only the bots with uncalculated measurements “wait” for user input before completing data storage. The new measurement rules will be used the next time bots are initialized in accordance with the frequency established by the user (20). In either event, system processing then passes on to a software block 269.

The software in block 269 checks the system settings table (140) to see if semantic mapping is being used. If semantic mapping is not being used, then processing advances to a block 281. Alternatively, if the software in block 269 determines that semantic mapping is being used, processing advances to a software block 270.

The software in block 270 checks the bot date table (149) and deactivates any inference bots with creation dates before the current system date and retrieves information from the system settings table (140), the metadata mapping table (141), the keyword table (150) and the classified text table (151). The software in block 270 then initializes inference bots for each keyword in the metadata mapping table (141) that mapped to the classified text table (151) in the application database (50) in accordance with the frequency specified by user (20) in the system settings table (140).

Bots are independent components of the application that have specific tasks to perform. In the case of inference bots, their task is to use Bayesian inference algorithms to determine the characteristics that give meaning to the text associated with keywords and classified text previously stored in the application database (50). Every inference bot contains the information shown in Table 21.

Tabl 21

1. Unique ID number (based on date, hour, minute, second of creation)
2. Creation date (date, hour, minute, second)
3. Mapping information
4. Storage location
5. Enterprise
6. Keyword
7. Classified text mapping information

After being activated, the inference bots determine the characteristics that give the text meaning in accordance with their programmed instructions with the frequency specified by the user (20) in the system settings table (140). The information defining the characteristics that give the text meaning is stored in the semantic map table (173) in the application database (50) before processing advances to block 272.

The software in block 272 checks the semantic map table (173) to see if there are new semantic maps. If there are no new semantic maps, then processing advances to a block 281. Alternatively, if the software in block 272 determines that there are new semantic maps, then processing returns to software block 255 and the processing described previously for Internet, text and geospatial bots is repeated.

The software in block 281 checks: the basic financial system table (143), the operation system table (144), the human resource system table (145), the external database table (146), the advanced finance system table (147), the asset system table (148), the classified text table (151), the geospatial measures table (152), the supply chain system table (174) and the risk system table (176) to see if data are missing from any of the periods required for system calculation. The software in block 202 previously calculated the range of required dates. If there are no data missing from any required period, then processing advances to a software block 283. Alternatively, if there are missing data for any field for any period, then processing advances to a block 282.

The software in block 282, prompts the user (20) via the missing data window (704) to specify the method to be used for filling the blanks for each item that is missing data. Options the user (20) can choose from for filling the blanks include: the average value for

the item over the entire time period, the average value for the item over a specified period, zero, the average of the preceding item and the following item values and direct user input for each missing item. If the user (20) does not provide input within a specified interval, then the default missing data procedure specified in the system settings table (140) is used. When all the blanks have been filled and stored for all of the missing data, system processing advances to a block 283.

The software in block 283 calculates attributes by item for each numeric item variable in the basic financial system table (143), the operation system table (144), the human resource system table (145), the external database table (146), the advanced finance system table (147), the asset system table (148), the supply chain system table (174) and the risk system table (176). The attributes calculated in this step include: summary data like cumulative total value; ratios like the period to period rate of change in value; trends like the rolling average value, comparisons to a baseline value like change from a prior years level and time shifted values like the time lagged value of each numeric item variable. The software in block 283 calculates similar attributes for the text and geospatial item variables created previously. The software in block 283 calculates attributes for each date item variable in the extracted text data and specified tables (143, 144, 145, 146, 147, 148, 174 and 176) including summary data like time since last occurrence and cumulative time since first occurrence; and trends like average frequency of occurrence and the rolling average frequency of occurrence. The numbers derived from the item variables are collectively referred to as "item performance indicators". The software in block 283 also calculates pre-specified combinations of variables called composite variables for measuring the strength of the different elements of value. The item performance indicators are stored in the table where the item source data was obtained and the composite variables are stored in the composite variables table (153) before processing advances to a block 284.

The software in block 284 uses attribute derivation algorithms such as the AQ program to create combinations of the variables that were not pre-specified for combination. While the AQ program is used in one embodiment of the present invention, other attribute derivation algorithms, such as the LINUS algorithms, may be used to the same effect. The software creates these attributes using both item variables that were specified as "element" variables and item variables that were not. The resulting composite variables

are stored in the composite variables table (153) before processing advances to a block 285.

The software in block 285 derives external factor indicators for each numeric data field defined in the external factor definition table (169). For example, external factors include: the ratio of organization earnings to expected earnings, the number and amount of jury awards, commodity prices, the inflation rate, growth in g.d.p., organization earnings volatility vs. industry average volatility, short and long term interest rates, increases in interest rates, insider trading direction and levels, industry concentration, consumer confidence and the unemployment rate that have an impact on the market price of the equity for an organization and/or an industry. The external factor indicators derived in this step include: summary data like cumulative totals, ratios like the period to period rate of change, trends like the rolling average value, comparisons to a baseline value like change from a prior years price and time lagged data like time lagged earnings forecasts. In a similar fashion the software in block 285 calculates external factors for each date field in the external factor definition table (169) including summary factors like time since last occurrence and cumulative time since first occurrence; time shifted factors and trends like average frequency of occurrence and the rolling average frequency of occurrence. The numbers derived from numeric and date fields are collectively referred to as "factor performance indicators". The software in block 285 also calculates pre-specified combinations of variables called composite factors for measuring the strength of the different external factors. The external factors, factor performance indicators and the composite factors are stored in the factor variables table (176) before processing advances to a block 286.

The software in block 286 uses attribute derivation algorithms, such as the Linus algorithm, to create combinations of the factors that were not pre-specified for combination. While the Linus algorithm is used in one embodiment of the present invention, other attribute derivation algorithms, such as the AQ program, may be used to the same effect. The software creates these attributes using both external factors that were included in "composite factors" and external factors that were not. The resulting composite variables are stored in the factor variables table (176) before processing advances to a block 287.

The software in block 287 uses pattern-matching algorithms to assign pre-designated data fields for different elements of value to pre-defined groups with numerical values. This type of analysis is useful in classifying purchasing patterns and/or communications patterns as “heavy”, “light”, “moderate” or “sporadic”. This analysis is also be used to classify web site activity and advertising patterns in a similar fashion. The numeric values associated with the classifications are item performance indicators. They are stored in the application database (50) table where the item variables or factor variables they are derived from are located before processing advances to a block 288.

The software in block 288 retrieves data from the metadata mapping table (141) and system settings table (140) as required to create and then stores detailed definitions for the segments of value and the pre-defined components of value for the current operation in the segment definition table (156) . As discussed previously, there are up to five segments of value per organization - current operation, real options, derivatives, excess financial assets and market sentiment. The current operation is further subdivided into: a revenue component of value that is not divided into sub-components, the expense value that is divided into five sub-components: the cost of raw materials, the cost of manufacture or delivery of service, the cost of selling, the cost of support and the cost of administration and the capital value that is divided into six sub-components: cash, non-cash financial assets, production equipment, other assets, financial liabilities and equity in one embodiment. Different subdivisions of the components of value can be used to the same effect. When data storage is complete, processing advances to a software block 291.

The software in block 291 checks the derivatives table (175) in the application database (50) to see if there are historical values for all the derivatives stored in the table. Because SFAS 133 is still not fully implemented, some companies may not have data regarding the value of their derivatives during a time period where data are required. If there are values stored for all required time periods, then processing advances to a software block 302 where the analysis of the extracted data is started. Alternatively, if there are periods when the value of one or more derivatives has not been stored, then processing advances to a software block 292. The software in block 292 retrieves the required data from the external database table (146), the external factors table and the derivatives table (175) as required to value each derivative using a risk neutral valuation method for the time period or time periods that are missing values. The algorithms used for this analysis

can include Quasi Monte Carlo, equivalent Martingale or wavelets. When the calculations are completed, the resulting values are stored in the derivatives table (175) and processing advances to a block 293.

The software in block 293 prompts the user (20) via the frame definition window (705) to specify frames for analysis. Frames are sub-sets of the organization that can be analyzed at the value driver level separately. For example, the user (20) may wish to examine value and risk by country, by division, by project, by action, by program or by manager. The software in block 293 saves the frame definitions the user (20) specifies in the frame definition table (165) in the application database (50) before processing advances to a software block 294.

The software in block 294 retrieves the segment, element and factor variables from the: basic financial system (143), human resource system table (145), external database table (146), advanced finance system (147), asset system table (148), keyword table (150), classified text table (151), geospatial measures table (152), composite variables table (153), supply chain system table (174), derivatives table (175), risk system table (176), event risk table (178), financial forecasts table (179) and factor variables table (176) as required to assign frame designations to every element and factor variable that was stored in the application database (50) in the prior processing steps in this stage (200) of processing. After storing the revised segment, element and factor variables records in the same table they were retrieved from in the application database (50), the software in the block retrieves the definitions from the element definition table (155), segment definition table (156) and external factor definition table (169), updates them to reflect the new frame definitions and saves them in the appropriate table before processing advances to a software block 295.

The software in block 295 checks the: basic financial system (143), human resource system table (145), external database table (146), advanced finance system (147), asset system table (148), keyword table (150), classified text table (151), geospatial measures table (152), composite variables table (153), supply chain system table (174), derivatives table (175), risk system table (176), event risk table (178), financial forecasts table (179) and factor variables table (176) to see if there are frame assignments for all segment, element and factor variables. If there are frame assignments for all variables, then processing advances to a software block 302 where the analysis of the extracted data is

started. Alternatively, if there are variables without frame assignments, then processing advances to a software block 296.

The software in block 296 retrieves variables from the basic financial system (143), human resource system table (145), external database table (146), advanced finance system (147), asset system table (148), keyword table (150), classified text table (151), geospatial measures table (152), composite variables table (153), supply chain system table (174), derivatives table (175), risk system table (176), event risk table (178), financial forecasts table (179) and factor variables table (176) that don't have frame assignments and then prompts the user (20) via the frame assignment window (705) to specify frame assignments for these variables. The software in block 296 saves the frame assignments the user (20) specifies as part of the data record for the variable in the table where the variable was retrieved from in the application database (50) before processing advances to software block 302 to begin the value analysis of the extracted data.

VALUE ANALYSIS

The flow diagrams in FIG. 6A, FIG. 6B and FIG. 6C detail the processing that is completed by the portion of the application software (300) that continually values the segments of value. This portion of the application software (300) also generates a matrix quantifying the impact of elements of value and external factors on the segments of value for the organization (see FIG. 8) by creating and activating analysis bots that:

- 1) Identify the factor variables, factor performance indicators and composite variables for each external factor that drive: three of the segments of value - current operation, derivatives and excess financial assets - as well as the components of current operation value (revenue, expense and changes in capital);
- 2) Identify the item variables, item performance indicators and composite variables for each element and sub-element of value that drive: three segments of value - current operation, derivatives and financial assets - as well as the components of current operation value (revenue, expense and changes in capital);
- 3) Create vectors that summarize the impact of the factor variables, factor performance indicators and composite variables for each external factor;

- 4) Create vectors that summarize the performance of the item variables, item performance indicators and composite variables for each element of value and sub-element of value in driving segment value;
- 5) Determine the expected life of each element of value and sub-element of value;
- 6) Determine the current operation value, excess financial asset value and derivative value, revenue component value, expense component value and capital component value of said current operations using the information prepared in the previous stages of processing;
- 7) Specify and optimize causal predictive models to determine the relationship between the vectors generated in steps 3 and 4 and the three segments of value, current operation, derivatives and financial assets, as well as the components of current operation value (revenue, expense and changes in capital);
- 8) Determine the appropriate discount rate on the basis of relative causal element strength, value the organization real options and contingent liabilities and determine the contribution of each element to real option valuation;
- 9) Determine the best causal indicator for organization stock price movement, calculate market sentiment and analyze the causes of market sentiment; and
- 10) Combine the results of all prior stages of processing to determine the value of each element, sub-element and factor for the organization.

Each analysis bot generally normalizes the data being analyzed before processing begins. While the processing in one embodiment includes an analysis of all five segments of value for the enterprise, it is to be understood that the system of the present invention can complete calculations for any combination of the five segments. For example, when a company is privately held it does not have a market price and as a result the market sentiment segment of value is not analyzed.

Processing in this portion of the application begins in software block 302. The software in block 302 checks the system settings table (140) in the application database (50) to determine if the current calculation is a new calculation or a structure change. If the calculation is not a new calculation or a structure change, then processing advances to a software block 315. Alternatively, if the calculation is new or a structure change, then processing advances to a software block 303.

The software in block 303 retrieves data from the system settings table (140), the meta data mapping table (141), the asset system table (148), the element definition table (155) and the frame definition table (165) and then assigns item variables, item performance indicators and composite variables to each element of value identified in the system settings table (140) using a three-step process. First, item variables, item performance indicators and composite variables are assigned to elements of value based on the asset management system they correspond to (for example, all item variables from a brand management system and all item performance indicators and composite variables derived from brand management system item variables are assigned to the brand element of value). Second, pre-defined composite variables are assigned to the element of value they were assigned to measure in the metadata mapping table (141). Finally, item variables, item performance indicators and composite variables identified by the text and geospatial bots are assigned to elements on the basis of their element classifications. If any item variables, item performance indicators or composite variables are un-assigned at this point they are assigned to a going concern element of value. After the assignment of variables and indicators to elements is complete, the resulting assignments are saved to the element definition table (155) and processing advances to a block 304.

The software in block 304 retrieves data from the meta data mapping table (141), the external factor definition table (169) and the frame definition table (165) and then assigns factor variables, factor performance indicators and composite factors to each external factor. Factor variables, factor performance indicators and composite factors identified by the text and geospatial bots are then assigned to factors on the basis of their factor classifications. The resulting assignments are saved to external factor definition table (169) and processing advances to a block 305.

The software in block 305 checks the system settings table (140) in the application database (50) to determine if the organization has a market sentiment segment. If there is market sentiment, then processing advances to a block 306. Alternatively, if there are no market prices for equity for the organization, then processing advances to a software block 308.

The software in block 306 checks the bot date table (149) and deactivates any market value indicator bots with creation dates before the current system date. The software in block 306 then initializes market value indicator bots in accordance with the frequency

specified by the user (20) in the system settings table (140). The bot retrieves the information from the system settings table (140), the metadata mapping table (141) and the external factor definition table (169) before saving the resulting information in the application database (50).

Bots are independent components of the application that have specific tasks to perform. In the case of market value indicator bots their primary task is to identify the best market value indicator (price, relative price, yield, first derivative of price change or second derivative of price change) for the time period being examined. The market value indicator bots select the best value indicator by grouping the S&P 500 using each of the five value indicators with a Kohonen neural network. The resulting clusters are then compared to the known groupings of the S&P 500. The market value indicator that produced the clusters that most closely match the know S&P 500 is selected as the market value indicator. Every market value indicator bot contains the information shown in Table 22.

Table 22

1. Unique ID number (based on date, hour, minute, second of creation)
2. Creation date (date, hour, minute, second)
3. Mapping information
4. Storage location
5. Organization

When bot in block 306 have identified and stored the best market value indicator in the external factor definition table (169), processing advances to a block 307.

The software in block 307 checks the bot date table (149) and deactivates any temporal clustering bots with creation dates before the current system date. The software in block 307 then initializes a bot in accordance with the frequency specified by the user (20) in the system settings table (140). The bot retrieves information from the system settings table (140), the metadata mapping table (141) and the external database table (146) as required and define regimes for the organization market value before saving the resulting cluster information in the application database (50).

Bots are independent components of the application that have specific tasks to perform. In the case of temporal clustering bots, their primary task is to segment the market price data using the market value indicator selected by the bot in block 306 into distinct time regimes that share similar characteristics. The temporal clustering bot assigns a unique identification (id) number to each “regime” it identifies and stores the unique id numbers in the cluster id table (157). Every time period with data are assigned to one of the regimes. The cluster id for each regime is saved in the data record for each element variable and factor variable in the table where it resides . After the regimes are identified, the element and factor variables for the organization are segmented into a number of regimes less than or equal to the maximum specified by the user (20) in the system settings table (140). The time periods are segmented for an enterprise with a market value using a competitive regression algorithm that identifies an overall, global model before splitting the data and creating new models for the data in each partition. If the error from the two models is greater than the error from the global model, then there is only one regime in the data. Alternatively, if the two models produce lower error than the global model, then a third model is created. If the error from three models is lower than from two models then a fourth model is added. The process continues until adding a new model does not improve accuracy. Other temporal clustering algorithms may be used to the same effect. Every temporal clustering bot contains the information shown in Table 23.

Table 23

1. Unique ID number (based on date, hour, minute, second of creation)
2. Creation date (date, hour, minute, second)
3. Mapping information
4. Storage location
5. Maximum number of clusters
6. Organization

When bots in block 307 have identified and stored regime assignments for all time periods with data processing advances to a software block 308.

The software in block 308 checks the bot date table (149) and deactivates any variable clustering bots with creation dates before the current system date. The software in block 308 then initializes bots as required for each element of value and external factor. The bots: activate in accordance with the frequency specified by the user (20) in the system settings table (140), retrieve the information from the system settings table (140), the metadata mapping table (141), the element definition table (155) and external factor definition table (169) as required and define segments for the element variables and factor variables before saving the resulting cluster information in the application database (50).

Bots are independent components of the application that have specific tasks to perform. In the case of variable clustering bots, their primary task is to segment the element variables and factor variables into distinct clusters that share similar characteristics. The clustering bot assigns a unique id number to each "cluster" it identifies and stores the unique id numbers in the cluster id table (157). Every item variable for every element of value is assigned to one of the unique clusters. The cluster id for each variable is saved in the data record for each variable in the table where it resides. In a similar fashion, every factor variable for every external factor is assigned to a unique cluster. The cluster id for each variable is saved in the data record for the factor variable. The item variables and factor variables are segmented into a number of clusters less than or equal to the maximum specified by the user (20) in the system settings table (140). The data are segmented using the "default" clustering algorithm the user (20) specified in the system settings table (140). The system of the present invention provides the user (20) with the choice of several clustering algorithms including: an unsupervised "Kohonen" neural network, neural network, decision tree, support vector method, K-nearest neighbor, expectation maximization (EM) and the segmental K-means algorithm. For algorithms that normally require the number of clusters to be specified, the bot will iterate the number of clusters until it finds the cleanest segmentation for the data. Every variable clustering bot contains the information shown in Table 24.

Tabl 24

1. Unique ID number (based on date, hour, minute, second of creation)
2. Creation date (date, hour, minute, second)
3. Mapping information
4. Storage location
5. Element of value, sub element of value or external factor
6. Clustering algorithm type
7. Enterprise
8. Maximum number of clusters
9. Variable 1
...to
9+n. Variable n

When bots in block 308 have identified and stored cluster assignments for the variables associated with each element of value, sub-element of value or external factor, processing advances to a software block 309.

The software in block 309 checks the bot date table (149) and deactivates any predictive model bots with creation dates before the current system date. The software in block 309 then retrieves the information from the system settings table (140), the metadata mapping table (141), the element definition table (155), the segment definition table (156) and the external factor definition table (169) as required to initialize predictive model bots for each component of value.

Bots are independent components of the application that have specific tasks to perform. In the case of predictive model bots, their primary task is to determine the relationship between the element and factor variables and the derivative segment of value, the excess financial asset segment of value and the current operation segment of value. The predictive model bots also determine the relationship between the element variables and factor variables components of current operation value and sub-components of current operation value. Predictive model bots are initialized for each component of value, sub-component of value, derivative segment and excess financial asset segment. They are also initialized for each cluster and regime of data in

accordance with the cluster and regime assignments specified by the bots in blocks 307 and 308 . A series of predictive model bots is initialized at this stage because it is impossible to know in advance which predictive model type will produce the “best” predictive model for the data from each commercial enterprise. The series for each model includes 12 predictive model bot types: neural network; CART; GARCH, projection pursuit regression; generalized additive model (GAM), redundant regression network; rough-set analysis, boosted Naïve Bayes Regression; MARS; linear regression; support vector method and stepwise regression. Additional predictive model types can be used to the same effect. The software in block 309 generates this series of predictive model bots for the organization as shown in Table 25.

Table 25

Predictive models level
<p>Organization:</p> <p>Variables* relationship to organization cash flow (revenue – expense + capital change)</p> <p>Variables* relationship to organization revenue component of value</p> <p>Variables* relationship to organization expense subcomponents of value</p> <p>Variables* relationship to organization capital change subcomponents of value</p> <p>Variables* relationship to organization derivative segment of value</p> <p>Variables* relationship to organization excess financial asset segment of value</p>
<p>Enterprise:</p> <p>Variables** relationship to enterprise cash flow (revenue – expense + capital change)</p> <p>Variables** relationship to enterprise revenue component of value</p> <p>Variables** relationship to enterprise expense subcomponents of value</p> <p>Variables* relationship to enterprise capital change subcomponents of value</p> <p>Variables* relationship to enterprise derivative segment of value</p> <p>Variables* relationship to enterprise excess financial asset segment of value</p>

* Variables = enterprise variables,

**Variables = element and factor variables, item performance indicators.

Every predictive model bot contains the information shown in Table 26.

Table 26

1. Unique ID number (based on date, hour, minute, second of creation)
2. Creation date (date, hour, minute, second)
3. Mapping information
4. Storage location
5. Enterprise or Organization
6. Global or Cluster (ID) and/or Regime (ID)
7. Segment (Derivative, Excess Financial Asset or Current Operation)
8. Element, sub-element or external factor
9. Predictive Model Type

After predictive model bots are initialized, the bots activate in accordance with the frequency specified by the user (20) in the system settings table (140). Once activated, the bots retrieve the required data from the appropriate table in the application database (50) and randomly partition the element or factor variables into a training set and a test set. The software in block 309 uses “bootstrapping” where the different training data sets are created by re-sampling with replacement from the original training set so data records may occur more than once. After the predictive model bots complete their training and testing, processing advances to a block 310.

The software in block 310 determines if clustering improved the accuracy of the predictive models generated by the bots in software block 309. The software in block 310 uses a variable selection algorithm such as stepwise regression (other types of variable selection algorithms can be used) to combine the results from the predictive model bot analyses for each type of analysis – with and without clustering - to determine the best set of variables for each type of analysis. The type of analysis having the smallest amount of error as measured by applying the mean squared error algorithm to the test data is given preference in determining the best set of variables for use in later analysis. There are four possible outcomes from this analysis as shown in Table 27.

Tabl 27

1. Best model has no clustering
2. Best model has temporal clustering, no variable clustering
3. Best model has variable clustering, no temporal clustering
4. Best model has temporal clustering and variable clustering

If the software in block 310 determines that clustering improves the accuracy of the predictive models for an enterprise or organization, then processing advances to a software block 313. Alternatively, if clustering does not improve the overall accuracy of the predictive models for an enterprise or organization, then processing advances to a software block 311.

The software in block 311 uses a variable selection algorithm such as stepwise regression (other types of variable selection algorithms can be used) to combine the results from the predictive model bot analyses for each model to determine the best set of variables for each model. The models having the smallest amount of error as measured by applying the mean squared error algorithm to the test data is given preference in determining the best set of variables. As a result of this processing, the best set of variables contain: the item variables, factor variables, item performance indicators, factor performance indications, composite variables and compoiste factors that correlate most strongly with changes in the three segments being analyzed and the three components of value. The best set of variables will hereinafter be referred to as the "value drivers". Eliminating low correlation factors from the initial configuration of the vector creation algorithms increases the efficiency of the next stage of system processing. Other error algorithms alone or in combination may be substituted for the mean squared error algorithm. After the best set of variables have been selected and stored in the element variables table (158) or factor variables table (176) for all models at all levels for the enterprise, the software in block 311 tests the independence of the value drivers at the enterprise, external factor, element and sub-element level before processing advances to a block 312.

The software in block 312 checks the bot date table (149) and deactivates any causal predictive model bots with creation dates before the current system date. The software in block 312 then retrieves the information from the system settings table (140), the

metadata mapping table (141), the segment definition table (156), the element variables table (158) and the factor variables table (176) as required to initialize causal predictive model bots for each element of value, sub-element of value and external factor in accordance with the frequency specified by the user (20) in the system settings table (140).

Bots are independent components of the application that have specific tasks to perform. In the case of causal predictive model bots, their primary task is to refine the element and factor variable selection to reflect only causal variables. (Note: these variables are summed together to value an element when they are interdependent). A series of causal predictive model bots are initialized at this stage because it is impossible to know in advance which causal predictive model will produce the “best” vector for the best fit variables from each model. The series for each model includes five causal predictive model bot types: Tetrad, MML, LaGrange, Bayesian and path analysis. The software in block 312 generates this series of causal predictive model bots for each set of variables stored in the element variables table (158) and factor variables table (176) in the previous stage in processing. Every causal predictive model bot activated in this block contains the information shown in Table 28.

Table 28

1. Unique ID number (based on date, hour, minute, second of creation)
2. Creation date (date, hour, minute, second)
3. Mapping information
4. Storage location
5. Component or subcomponent of value
6. Element, sub-element or external factor
7. Variable set
8. Causal predictive model type
9. Enterprise or Organization

After the causal predictive model bots are initialized by the software in block 312, the bots activate in accordance with the frequency specified by the user (20) in the system settings table (140). Once activated, they retrieve the required information for each

model and sub-divide the variables into two sets, one for training and one for testing. After the causal predictive model bots complete their processing for each model, the software in block 312 uses a model selection algorithm to identify the model that best fits the data for each element of value, sub-element of value and external factor being analyzed. For the system of the present invention, a cross validation algorithm is used for model selection. The software in block 312 saves the best fit causal factors in the vector table (159) in the application database (50) and processing advances to a block 318.

The software in block 318 tests the value drivers to see if there is interaction between elements, between elements and external factors or between external factors by enterprise. The software in this block identifies interaction by evaluating a chosen model based on stochastic-driven pairs of value-driver subsets. If the accuracy of such a model is higher than the accuracy of statistically combined models trained on attribute subsets, then the attributes from subsets are considered to be interacting and then they form an interacting set. If the software in block 318 does not detect any value driver interaction or missing variables for the enterprise, then system processing advances to a block 323. Alternatively, if missing data or value driver interactions across elements are detected by the software in block 318 for one or more enterprise, then processing advances to a software block 321.

If software in block 310 determines that clustering improves predictive model accuracy, then processing advances to block 313 as described previously. The software in block 313 uses a variable selection algorithm such as stepwise regression (other types of variable selection algorithms can be used) to combine the results from the predictive model bot analyses for each model, cluster and/or regime to determine the best set of variables for each model. The models having the smallest amount of error as measured by applying the mean squared error algorithm to the test data is given preference in determining the best set of variables. As a result of this processing, the best set of variables contains: the element variables and factor variables that correlate most strongly with changes in the components of value. The best set of variables will hereinafter be referred to as the "value drivers". Eliminating low correlation factors from the initial configuration of the vector creation algorithms increases the efficiency of the next stage of system processing. Other error algorithms alone or in combination may be substituted for the mean squared error algorithm. After the best set of variables have been selected and stored in the element variables table (158) or the factor variables table (176) for all

models at all levels, the software in block 313 tests the independence of the value drivers at the enterprise, element, sub-element and external factor level before processing advances to a block 314.

The software in block 314 checks the bot date table (149) and deactivates any causal predictive model bots with creation dates before the current system date. The software in block 314 then retrieves the information from the system settings table (140), the metadata mapping table (141), the segment definition table (156), the element variables table (158) and the factor variables table (176) as required to initialize causal predictive model bots for each element of value, sub-element of value and external factor at every level in accordance with the frequency specified by the user (20) in the system settings table (140).

Bots are independent components of the application that have specific tasks to perform. In the case of causal predictive model bots, their primary task is to refine the element and factor variable selection to reflect only causal variables. (Note: these variables are grouped together to represent a single element vector when they are dependent). In some cases it may be possible to skip the correlation step before selecting causal the item variables, factor variables, item performance indicators, factor performance indicators, composite variables and composite factors. A series of causal predictive model bots are initialized at this stage because it is impossible to know in advance which causal predictive model will produce the "best" vector for the best fit variables from each model. The series for each model includes four causal predictive model bot types: Tetrad, LaGrange, Bayesian and path analysis. The software in block 314 generates this series of causal predictive model bots for each set of variables stored in the element variables table (158) in the previous stage in processing. Every causal predictive model bot activated in this block contains the information shown in Table 29.

Tabl 29

1. Unique ID number (based on date, hour, minute, second of creation)
2. Creation date (date, hour, minute, second)
3. Mapping information
4. Storage location
5. Component or subcomponent of value
6. Cluster (ID) and/or Regime (ID)
7. Element, sub-element or external factor
8. Variable set
9. Enterprise or Organization
10. Causal predictive model type

After the causal predictive model bots are initialized by the software in block 314, the bots activate in accordance with the frequency specified by the user (20) in the system settings table (140). Once activated, they retrieve the required information for each model and sub-divide the variables into two sets, one for training and one for testing. The same set of training data is used by each of the different types of bots for each model. After the causal predictive model bots complete their processing for each model, the software in block 314 uses a model selection algorithm to identify the model that best fits the data for each element, sub-element or external factor being analyzed by model and/or regime. For the system of the present invention, a cross validation algorithm is used for model selection. The software in block 314 saves the best fit causal factors in the vector table (159) in the application database (50) and processing advances to block 318. The software in block 318 tests the value drivers to see if there are "missing" value drivers that are influencing the results as well as testing to see if there are interactions (dependencies) across elements. If the software in block 318 does not detect any missing data or value driver interactions across elements, then system processing advances to a block 323. Alternatively, if missing data or value driver interactions across elements are detected by the software in block 318, then processing advances to a software block 321.

The software in block 321 prompts the user (20) via the structure revision window (710) to adjust the specification(s) for the affected elements of value, sub-elements of value or

external factors as required to minimize or eliminate the interaction. At this point the user (20) has the option of specifying that one or more elements of value, sub elements of value and/or external factors be combined for analysis purposes (element combinations and/or factor combinations) for the enterprise where there is interaction between elements and/or factors. The same process can be used when two or more enterprises share value drivers. The user (20) also has the option of specifying that the elements, external factors or enterprises that are interacting will be valued by summing the impact of their value drivers. Finally, the user (20) can chose to re-assign a value driver to a new element of value to eliminate the inter-dependency. This is one solution when the inter-dependent value driver is included in the going concern element of value. Elements and external factors that will be valued by summing their value drivers will not have vectors generated. After the input from the user (20) is saved in the system settings table (140), the element definition table (155) and the external factor definition table (169) system processing advances to a software block 323. The software in block 323 checks the system settings table (140), the element definition table (155) and/or the external factor definition table (169) to see if there any changes in structure. If there have been changes in the structure, then processing advances to a block 205 and the system processing described previously is repeated. Alternatively, if there are no changes in structure, then processing advances to a block 325.

The software in block 325 checks the system settings table (140) in the application database (50) to determine if the current calculation is a new one. If the calculation is new, then processing advances to a software block 326. Alternatively, if the calculation is not a new calculation, then processing advances to a software block 333.

The software in block 326 checks the bot date table (149) and deactivates any industry rank bots with creation dates before the current system date. The software in block 326 then retrieves the information from the system settings table (140), the metadata mapping table (141), and the vector table (159) as required to initialize industry rank bots for the enterprise and for the industry in accordance with the frequency specified by the user (20) in the system settings table (140).

Bots are independent components of the application that have specific tasks to perform. In the case of industry rank bots, their primary task is to determine the relative position of the enterprise being evaluated on element variables identified in the previous

processing step. (Note: these variables are grouped together when they are interdependent). The industry rank bots use ranking algorithms such as Data Envelopment Analysis (hereinafter, DEA) to determine the relative industry ranking of the element of value or enterprise being examined. The software in block 326 generates industry rank bots for the enterprise being evaluated. Every industry rank bot activated in this block contains the information shown in Table 30.

Table 30

1. Unique ID number (based on date, hour, minute, second of creation)
2. Creation date (date, hour, minute, second)
3. Mapping information
4. Storage location
5. Ranking algorithm
6. Element or Enterprise

After the industry rank bots are initialized by the software in block 326, the bots activate in accordance with the frequency specified by the user (20) in the system settings table (140). Once activated, they retrieve the item variables, item performance indicators, and composite variables from the application database (50) and sub-divides them into two sets, one for training and one for testing. After the industry rank bots develop and test their rankings, the software in block 326 saves the industry rankings in the vector table (159) in the application database (50) and processing advances to a block 327. The industry rankings are item variables.

The software in block 327 checks the bot date table (149) and deactivates any vector generation bots with creation dates before the current system date. The software in block 327 then initializes bots for each element of value, sub-element of value and external factor for the enterprise. The bots activate in accordance with the frequency specified by the user (20) in the system settings table (140), retrieve the information from the system settings table (140), the metadata mapping table (141), the segment definition table (156) and the element variables table (158) as required to initialize vector generation bots for each element of value and sub-element of value in accordance with the frequency specified by the user (20) in the system settings table (140).

Bots are independent components of the application that have specific tasks to perform. In the case of vector generation bots, their primary task is to produce formulas, (hereinafter, vectors) that summarize the relationship between the causal element variables or causal factor variables and changes in the component or sub-component of value being examined for the enterprise. The causal element variables may be grouped by element of value, sub-element of value, external factor, factor combination or element combination. As discussed previously, the vector generation step is skipped for elements and factors where the user has specified that value driver impacts will be mathematically summed to determine the value of the element or factor. The vector generation bots use induction algorithms to generate the vectors. Other vector generation algorithms can be used to the same effect. The software in block 327 generates a vector generation bot for each set of variables stored in the element variables table (158) and factor variables table (176). Every vector generation bot contains the information shown in Table 31.

Table 31

1. Unique ID number (based on date, hour, minute, second of creation)
2. Creation date (date, hour, minute, second)
3. Mapping information
4. Storage location
5. Enterprise
6. Element, sub-element, element combination, factor or factor combination
7. Component or sub-component of value
8. Factor 1
...to
8+n. Factor n

When bots in block 327 have identified and stored vectors for all time periods with data for all the elements, sub-elements, element combination, factor combination or external factor where vectors are being calculated in the vector table (163), processing advances to a software block 329.

The software in block 329 checks the bot date table (149) and deactivates any financial factor bots with creation dates before the current system date. The software in block 329

then retrieves the information from the system settings table (140), the metadata mapping table (141), the element definition table (155), the element variables table (158), the external factor definition table (169), the derivatives table (175), the financial forecasts table (179) and the factor variables table (176) as required to initialize causal external factor bots for the enterprise and the relevant industry in accordance with the frequency specified by the user (20) in the system settings table (140).

Bots are independent components of the application that have specific tasks to perform. In the case of financial factor bots, their primary task is to identify elements of value, value drivers and external factors that are causal factors for changes in the value of: derivatives, financial assets, enterprise equity and industry equity. The causal factors for organization equity and industry equity are those that drive changes in the value indicator identified by the value indicator bots. A series of financial factor bots are initialized at this stage because it is impossible to know in advance which causal factors will produce the "best" model for every derivative, financial asset, enterprise or industry. The series for each model includes five causal predictive model bot types: Tetrad, LaGrange, MML, Bayesian and path analysis. Other causal predictive models can be used to the same effect. The software in block 329 generates this series of causal predictive model bots for each set of variables stored in the element variables table (158) and factor variables table (176) in the previous stage in processing. Every financial factor bot activated in this block contains the information shown in Table 32

Table 32

1. Unique ID number (based on date, hour, minute, second of creation)
2. Creation date (date, hour, minute, second)
3. Mapping information
4. Storage location
5. Element, value driver or external factor
6. Enterprise or Organization
7. Type: derivatives, financial assets, enterprise equity or industry equity
8. Value indicator (price, relative price, first derivative, etc.)
9. Causal predictive model type

After the software in block 329 initializes the financial factor bots, the bots activate in accordance with the frequency specified by the user (20) in the system settings table (140). Once activated, they retrieve the required information and sub-divide the data into two sets, one for training and one for testing. The same set of training data is used by each of the different types of bots for each model. After the financial factor bots complete their processing for each segment of value, organization and industry, the software in block 329 uses a model selection algorithm to identify the model that best fits the data for each. For the system of the present invention, a cross validation algorithm is used for model selection. The software in block 329 saves the best fit causal factors in the factor variables table (176) and the best fit causal elements and value drivers in the element variables table (158) by enterprise and processing advances to a block 330. The software in block 330 tests to see if there are "missing" causal factors, elements or value drivers that are influencing the results. If the software in block 330 does not detect any missing factors, elements or value drivers, then system processing advances to a block 331. Alternatively, if missing factors, elements or value drivers are detected by the software in block 330, then processing returns to software block 321 and the processing described in the preceding section is repeated.

The software in block 331 checks the bot date table (149) and deactivates any option bots with creation dates before the current system date. The software in block 331 then retrieves the information from the system settings table (140), the metadata mapping table (141), the basic financial system database (143), the external database table (146), the advanced finance system table (147) and the vector table (163) as required to initialize option bots for the enterprise.

Bots are independent components of the application that have specific tasks to perform. In the case of option bots, their primary tasks are to calculate the discount rate to be used for valuing the real options and contingent liabilities and to value the real options and contingent liabilities for the enterprise. If the user (20) has chosen to include industry options, then option bots will be initialized for industry options as well. The discount rate for enterprise real options is calculated by adding risk factors for each causal element to a base discount rate. A two step process determines the risk factor for each causal element. The first step in the process divides the maximum real option discount rate (specified by the user in system settings) by the number of causal elements. The second step in the process determines if the enterprise is highly rated on the causal

elements using ranking algorithms like DEA and determines an appropriate risk factor. If the enterprise is highly ranked on the soft asset, then the discount rate is increased by a relatively small amount for that causal element. Alternatively, if the enterprise has a low ranking on a causal element, then the discount rate is increased by a relatively large amount for that causal element as shown below in Table 33.

Table 33

Maximum discount rate = 50%, Causal elements = 5		
Maximum risk factor/soft asset = 50%/5= 10%		
Industry Rank on Soft Asset		% of Maximum
1		0%
2		25%
3		50%
4		75%
5 or higher		100%
Causal element:	Relative Rank	Risk Factor
Brand	1	0%
Channel	3	5%
Manufacturing Process	4	7.5%
Strategic Alliances	5	10%
Vendors	2	2.5%
Subtotal		25%
Base Rate		12%
Discount Rate		37%

The discount rate for industry options is calculated using a traditional total cost of capital approach that includes the cost of risk capital in a manner that is well known. After the appropriate discount rates are determined, the value of each real option and contingent liability is calculated using the specified algorithms in a manner that is well known. The real option can be valued using a number of algorithms including Black Scholes, binomial, neural network or dynamic programming algorithms. The industry option bots use the industry rankings from prior processing block to determine an allocation percentage for industry options. The more dominant the enterprise, as indicated by the industry rank for the element indicators, the greater the allocation of industry real options. Every option bot contains the information shown in Table 34.

Table 34

1. Unique ID number (based on date, hour, minute, second of creation)
2. Creation date (date, hour, minute, second)
3. Mapping information
4. Storage location
5. Industry or Enterprise
6. Real option type (Industry or Enterprise)
7. Real option algorithm (Black Scholes, Binomial, Quadrantomial, Dynamic Program, etc.)

After the option bots are initialized, they activate in accordance with the frequency specified by the user (20) in the system settings table (140). After being activated, the bots retrieve information as required to complete the option valuations. When they are used, industry option bots go on to allocate a percentage of the calculated value of industry options to the enterprise on the basis of causal element strength. After the value of the real option, contingent liability or allocated industry option is calculated the resulting values are then saved in the real option value table (162) in the application database (50) before processing advances to a block 332.

The software in block 332 checks the bot date table (149) and deactivates any cash flow bots with creation dates before the current system date. The software in the block then retrieves the information from the system settings table (140), the metadata mapping table (141), the advanced finance system table (147) and the segment definition table (156) as required to initialize cash flow bots for the enterprise in accordance with the frequency specified by the user (20) in the system settings table (140).

Bots are independent components of the application that have specific tasks to perform. In the case of cash flow bots, their primary tasks are to calculate the cash flow for the enterprises and organization for every time period where data are available and to forecast a steady state cash flow for the enterprise. Cash flow is calculated using the forecast revenue, expense, capital change and depreciation data retrieved from the advanced finance system table (147) with a well-known formula where cash flow equals period revenue minus period expense plus the period change in capital plus non-cash depreciation/amortization for the period. The steady state cash flow for the enterprise is

calculated for the enterprise using forecasting methods identical to those disclosed previously in U.S. Patent 5,615,109 to forecast revenue, expenses, capital changes and depreciation separately before calculating the cash flow. Every cash flow bot contains the information shown in Table 35.

Table 35

1. Unique ID number (based on date, hour, minute, second of creation)
2. Creation date (date, hour, minute, second)
3. Mapping information
4. Storage location
5. Enterprise or Organization

After the cash flow bots are initialized, the bots activate in accordance with the frequency specified by the user (20) in the system settings table (140). After being activated the bots, retrieve the forecast data for the enterprise from the advanced finance system table (147) and then calculate a steady state cash flow forecast. The resulting values by period for the enterprise are then stored in the cash flow table (161) in the application database (50) before processing advances to a block 333.

The software in block 333 checks the system settings table (140) in the application database (50) to determine if the current calculation is a new calculation or a structure change. If the calculation is not a new calculation or a structure change, then processing advances to a software block 341. Alternatively, if the calculation is new or a structure change, then processing advances to a software block 343.

The software in block 341 uses the cash flow by period data from the cash flow table (161) and the calculated requirement for working capital to calculate the value of excess financial assets for every time period and stores the results of the calculation in the financial forecasts table (179) in the application database before processing advances to a block 342.

The software in block 342 checks the bot date table (149) and deactivates any financial value bots with creation dates before the current system date. The software in block 342 then retrieves the information from the system settings table (140), the metadata mapping

table (141), the element definition table (155), the element variables table (158), the external factor definition table (169), the derivatives table (175) the financial forecasts table (179) and the factor variables table (176) as required to initialize financial value bots for the derivatives and excess financial assets in accordance with the frequency specified by the user (20) in the system settings table (140).

Bots are independent components of the application that have specific tasks to perform. In the case of financial value bots, their primary task is to determine the relative contribution of element data and factor data identified in previous stages of processing on the value of derivatives and excess financial assets. The system of the present invention uses 12 different types of predictive models to determine relative contribution: neural network; CART; projection pursuit regression; generalized additive model (GAM); GARCH; MMDR; redundant regression network; boosted Naïve Bayes Regression; the support vector method; MARS; linear regression; and stepwise regression. The model having the smallest amount of error as measured by applying the mean squared error algorithm to the test data is the best fit model. The "relative contribution algorithm" used for completing the analysis varies with the model that was selected as the "best-fit" as described previously. Every financial value bot activated in this block contains the information shown in Table 36.

Table 36

1. Unique ID number (based on date, hour, minute, second of creation)
2. Creation date (date, hour, minute, second)
3. Mapping information
4. Storage location
5. Enterprise or Organization
6. Derivative or Excess Financial Asset
7. Element Data or Factor Data
8. Predictive model type

After the software in block 342 initializes the financial value bots, the bots activate in accordance with the frequency specified by the user (20) in the system settings table (140). Once activated, they retrieve the required information and sub-divide the data into

two sets, one for training and one for testing. The same set of training data is used by each of the different types of bots for each model. After the financial bots complete their processing, the software in block 332 saves the calculated value contributions by element or external factor for derivatives in the derivatives table (175) . The calculated value contributions by element or external factor for excess financial assets are then saved in the financial forecasts table (179) in the application database (50) and processing advances to a block 343.

The software in block 343 checks the bot date table (149) and deactivates any element life bots with creation dates before the current system date. The software in block 343 then retrieves the information from the system settings table (140), the metadata mapping table (141) and the element definition table (155) as required to initialize element life bots for each element and sub-element of value for the enterprise being analyzed.

Bots are independent components of the application that have specific tasks to perform. In the case of element life bots, their primary task is to determine the expected life of each element and sub-element of value. There are three methods for evaluating the expected life of the elements and sub-elements of value. Elements of value that are defined by a population of members or items (such as: channel partners, customers, employees and vendors) will have their lives estimated by analyzing and forecasting the lives of the members of the population. The forecasting of member lives will be determined by the “best” fit solution from competing life estimation methods including the Iowa type survivor curves, Weibull distribution survivor curves, Gompertz-Makeham survivor curves, polynomial equations using the methodology for selecting from competing forecasts disclosed in U.S. Patent 5,615,109. Elements of value (such as some parts of Intellectual Property i.e. patents and insurance contracts) that have legally defined lives will have their lives calculated using the time period between the current date and the expiration date of the element or sub-element. Finally, elements of value and sub-element of value (such as brand names, information technology and processes) that may not have defined lives and/or that may not consist of a collection of members will have their lives estimated as a function of the enterprise Competitive Advantage Period (CAP). In the latter case, the estimate will be completed using the element vector trends and the stability of relative element strength. More specifically, lives for these element types are estimated by

- 1) subtracting time from the CAP for element volatility that exceeds cap volatility; and/or
- 2) subtracting time for relative element strength that is below the leading position and/or relative element strength that is declining;

The resulting values are stored in the element definition table (155) for each element and sub-element of value . Every element life bot contains the information shown in Table 37.

Table 37

1. Unique ID number (based on date, hour, minute, second of creation)
2. Creation date (date, hour, minute, second)
3. Mapping information
4. Storage location
5. Enterprise or Organization
6. Element or sub-element of value
7. Life estimation method (item analysis, date calculation or relative to CAP)

After the element life bots are initialized, they are activated in accordance with the frequency specified by the user (20) in the system settings table (140). After being activated, the bots retrieve information for each element and sub-element of value from the element definition table (155) as required to complete the estimate of element life. The resulting values are then saved in the element definition table (155) in the application database (50) before processing advances to a block 345.

The software in block 345 checks the system settings table (140) in the application database (50) to determine if the current calculation is a new calculation or a structure change. If the calculation is not a new calculation or a structure change, then processing advances to a software block 402. Alternatively, if the calculation is new or a structure change, then processing advances to a software block 348.

The software in block 348 checks the bot date table (149) and deactivates any component capitalization bots with creation dates before the current system date. The software in block 348 then retrieves the information from the system settings table (140),

the metadata mapping table (141) and the segment definition table (156) as required to initialize component capitalization bots for the enterprise.

Bots are independent components of the application that have specific tasks to perform. In the case of component capitalization bots, their task is to determine the capitalized value of the components and subcomponents of value - forecast revenue, forecast expense or forecast changes in capital for the enterprise in accordance with the formula shown in Table 38.

Table 38

$$\text{Value} = F_{t1} / (1+K) + F_{t2} / (1+K)^2 + F_{t3} / (1+K)^3 + F_{t4} / (1+K)^4 + (F_{t4} \times (1+g)) / (1+K)^5 + (F_{t4} \times (1+g)^2) / (1+K)^6 \dots + (F_{t4} \times (1+g)^N) / (1+K)^{N+4}$$

Where:

F_{tx}	= Forecast revenue, expense or capital requirements for year x after valuation date (from advanced finance system)
N	= Number of years in CAP (from prior calculation)
K	= Total average cost of capital - % per year (from prior calculation)
g	= Forecast growth rate during CAP - % per year (from advanced financial system)

After the calculation of capitalized value of every component and sub-component of value is complete, the results are stored in the segment definition table (156) in the application database (50). Every component capitalization bot contains the information shown in Table 39.

Table 39

1. Unique ID number (based on date, hour, minute, second of creation)
2. Creation date (date, hour, minute, second)
3. Mapping information
4. Storage location
5. Enterprise
6. Component of value (revenue, expense or capital change)
7. Sub component of value

After the component capitalization bots are initialized, they activate in accordance with the frequency specified by the user (20) in the system settings table (140). After being activated, the bots retrieve information for each component and sub-component of value from the advanced finance system table (147) and the segment definition table (156) as required to calculate the capitalized value of each component for the enterprise. The resulting values are then saved in the segment definition table (156) in the application database (50) before processing advances to a block 349.

The software in block 349 checks the bot date table (149) and deactivates any current operation bots with creation dates before the current system date. The software in block 349 then retrieves the information from the system settings table (140), the metadata mapping table (141), the element definition table (155), the segment definition table (156), the vector table (163), the external factor definition table (169), the financial forecasts table (179) and the factor variables table (176) as required to initialize valuation bots for each element of value, sub-element of value, combination of elements, value driver and/or external factor for the current operation.

Bots are independent components of the application that have specific tasks to perform. In the case of current operation bots, their task is to calculate the contribution of every element of value, sub-element of value, element combination, value driver, external factor and factor combination to the current operation segment of enterprise value. For calculating the current operation portion of element value, the bots use the procedure outlined in Table 5. The first step in completing the calculation in accordance with the procedure outlined in Table 5, is determining the relative contribution of each element, sub-element, combination of elements or value driver by using a series of predictive models to find the best fit relationship between:

1. The element of value vectors, element combination vectors and external factor vectors, factor combination vectors and value drivers and the enterprise components of value they correspond to; and
2. The sub-element of value vectors and the element of value they correspond to.

The system of the present invention uses 12 different types of predictive models to identify the best fit relationship: neural network; CART; projection pursuit regression; generalized additive model (GAM); GARCH; MMDR; redundant regression network;

boosted Naïve Bayes Regression; the support vector method; MARS; linear regression; and stepwise regression. The model having the smallest amount of error as measured by applying the mean squared error algorithm to the test data is the best fit model. The “relative contribution algorithm” used for completing the analysis varies with the model that was selected as the “best-fit”. For example, if the “best-fit” model is a neural net model, then the portion of revenue attributable to each input vector is determined by the formula shown in Table 40.

Table 40

$\left(\sum_{k=1}^{k=m} \sum_{j=1}^{j=n} I_{jk} \times O_k / \sum_{j=1}^{j=n} I_{ik} \right) / \sum_{k=1}^{k=m} \sum_{j=1}^{j=n} I_{jk} \times O_k$			
Where			
I_{jk}	= Absolute value of the input weight from input node j to hidden node k		
O_k	= Absolute value of output weight from hidden node k		
M	= number of hidden nodes		
N	= number of input nodes		

After the relative contribution of each element of value, sub-element of value, external factor, element combination, factor combination and value driver to the components of current operation value is determined, the results of this analysis are combined with the previously calculated information regarding element life and capitalized component value to complete the valuation of each: element of value, sub-element of value, external factor, element combination, factor combination and value driver using the approach shown in Table 41.

Table 41

Component Values:	Percentage	Element Life/CAP	Net Value
Revenue value = \$120M	20%	80%	Value = \$19.2 M
Expense value = (\$80M)	10%	80%	Value = (\$6.4) M
Capital value = (\$5M)	5%	80%	Value = (\$0.2) M
Total value = \$35M			
Net value for this element:			Value = \$12.6 M

The resulting values are stored in: the element definition table (155) for each element of value, sub-element of value, element combination and value driver . For external factor and factor combination value calculations, the external factor percentage is multiplied by the capitalized component value to determine the external factor value. The resulting values for external factors are saved in the external factor definition table (169) .

Every current operation bot contains the information shown in Table 42.

Table 42

1. Unique ID number (based on date, hour, minute, second of creation)
2. Creation date (date, hour, minute, second)
3. Mapping information
4. Storage location
5. Enterprise
6. Element, sub-element, factor, element combination, factor combination or value driver
7. Component of value (revenue, expense or capital change)

After the current operation bots are initialized by the software in block 349 they activate in accordance with the frequency specified by the user (20) in the system settings table (140). After being activated, the bots retrieve information and complete the valuation for the segment being analyzed. As described previously, the resulting values are then saved in the element definition table (155) or the external factor definition table (169) in the application database (50) before processing advances to a block 350.

The software in block 350 checks the bot date table (149) and deactivates any residual bots with creation dates before the current system date. The software in block 350 then retrieves the information from the system settings table (140), the metadata mapping table (141), the element definition table (155) and the external factor definition table (169) as required to initialize residual bots for the the enterprise.

Bots are independent components of the application that have specific tasks to perform. In the case of residual bots, their task is to retrieve data as required from the element definition table (155) the segment definition table (156) and the external factor

definition table (169) to calculate the residual going concern value for the enterprise in accordance with the formula shown in Table 43.

Table 43

$\text{Residual Going Concern Value} = \text{Total Current-Operation Value} - \sum \text{Required Financial Asset Values} - \sum \text{Elements of Value} - \sum \text{External Factors}$

Every residual bot contains the information shown in Table 44.

Table 44

1. Unique ID number (based on date, hour, minute, second of creation)
2. Creation date (date, hour, minute, second)
3. Mapping information
4. Storage location
5. Enterprise

After the residual bots are initialized they activate in accordance with the frequency specified by the user (20) in the system settings table (140). After being activated, the bots retrieve information as required to complete the residual calculation for the enterprise. After the calculation is complete, the resulting values are then saved in the element definition table (155) in the application database (50) before processing advances to a software block 351.

The software in block 351 determines the contribution of each element of value to the value of the real option segment of value for the enterprise. For enterprise options, the value of each element is determined by comparing the value of the enterprise options to the value that would have been calculated if the element had an average level of strength. Elements that are relatively strong, reduce the discount rate and increase the value of the option. In a similar fashion, elements that are below average in strength increase the discount rate and decrease the value of the option. The value impact can be determined by subtracting the calculated value of the option from the value of the option with the average element. The resulting values are saved in the element definition table (155) before processing advances to block 352.

The software in block 352 checks the bot date table (149) and deactivates any sentiment calculation bots with creation dates before the current system date. The software in block 352 then retrieves the information from the system settings table (140), the metadata mapping table (141), the external database table (146), the element definition table (155), the segment definition table (156), the real option value table (162) and the derivatives table (175) as required to initialize sentiment calculation bots for the enterprise.

Bots are independent components of the application that have specific tasks to perform. In the case of sentiment calculation bots, their task is to retrieve data as required and then calculate the sentiment for the enterprise in accordance with the formula shown in Table 45.

Table 45

$\text{Sentiment} = \text{Market Value for Enterprise} - \text{Current Operation Value} - \sum \text{Real Option Values} - \text{Value of Excess Financial Assets} - \sum \text{Derivative Values}$

If the enterprise is not a public corporation be no market sentiment calculation. Every sentiment calculation bot contains the information shown in Table 46.

Table 46

1. Unique ID number (based on date, hour, minute, second of creation)
2. Creation date (date, hour, minute, second)
3. Mapping information
4. Storage location
5. Enterprise

After the sentiment calculation bots are initialized, they activate in accordance with the frequency specified by the user (20) in the system settings table (140). After being activated, the bots retrieve information from the system settings table (140), the external database table (146), the element definition table (155), the segment definition table (156), the real option value table (162), the derivatives table (175) and the financial forecasts table (179) as required to complete the sentiment calculation for the enterprise. After the calculation is complete, the resulting values are then saved in the market

sentiment table (166) in the application database (50) before processing advances to a block 353.

The software in block 353 checks the bot date table (149) and deactivates any sentiment analysis bots with creation dates before the current system date. The software in block 352 then retrieves the information from the system settings table (140), the metadata mapping table (141), the external database table (146), the industry ranking table (154), the element definition table (155), the segment definition table (156), the real option value table (162), the vector table (163), the market sentiment table (166) and the external factor definition table (169) as required to initialize sentiment analysis bots for the enterprise.

Bots are independent components of the application that have specific tasks to perform. In the case of sentiment analysis bots, their primary task is to determine the composition of the calculated sentiment for the enterprise. One part of this analysis is completed by comparing the portion of overall market value that is driven by the different elements of value as determined by the bots in software block 329 and the calculated valuation impact of each element of value on the segments of value as shown below in Table 47.

Table 47

Total Organization Market Value = \$100 Billion, 10% driven by Brand factors
Implied Brand Value = \$100 Billion X 10% = \$10 Billion
Brand Element Current Operation Value = \$6 Billion
Increase/(Decrease) in Enterprise Real Option Values* Due to Brand = \$1.5 Billion
Increase/(Decrease) in Derivative Values due to Brands = \$0.0
Increase/(Decrease) in excess Financial Asset Values due to Brands = \$0.25 Billion
Brand Sentiment = \$10 - \$6 - \$1.5 - \$0.0 - \$0.25 = \$2.25 Billion

* includes allocated industry options when used in the calculation

The sentiment analysis bots also determine the impact of external factors on sentiment. Every sentiment analysis bot contains the information shown in Table 48.

Tabl 48

1. Unique ID number (based on date, hour, minute, second of creation)
2. Creation date (date, hour, minute, second)
3. Mapping information
4. Storage location
5. External factor or element of value
6. Enterprise

After the sentiment analysis bots are initialized, they activate in accordance with the frequency specified by the user (20) in the system settings table (140). After being activated, the bots retrieve information from the system settings table (140), the metadata mapping table (141), the industry ranking table (154), the element definition table (155), the segment definition table (156), the real option value table (162), the market sentiment table (166), the external factor definition table (169), the derivatives table (175) and the financial forecasts table (179) as required to analyze sentiment. The resulting breakdown of sentiment is then saved in the enterprise sentiment table (169) in the application database (50) before processing advances to a software block 402 where the management reports are created and the frames for platform analysis are defined.

ANALYSIS & OUTPUT

The flow diagram in FIG. 7 details the processing that is completed by the portion of the application software (400) that generates the value management matrix for the enterprise, generates a summary of the value, risk and liquidity for the enterprise, analyzes changes in enterprise structure and operation and optionally displays and prints management reports detailing the value matrix, risk matrix and the efficient frontier. Processing in this portion of the application starts in software block 402.

The software in block 402 retrieves information from the system settings table (140), the advanced finance system table the cash flow table (161) and the financial forecasts table (179) that is required to calculate the minimum amount of working capital that will be available during the forecast time period. The system settings table (140) contains the minimum amount of working capital that the user (20) indicated was required for enterprise operation while the cash flow table (161) contains a forecast of the cash flow of

the enterprise for each period during the forecast time period (generally the next 36 months). A summary of the available cash and cash deficits by currency, by month, for the next 36 months is stored in a summary xml format in the xml summary table (177) during this stage of processing. After the amount of available cash for the enterprise is calculated and stored in the risk reduction purchase table (165), processing advances to a software block 403.

The software in block 403 retrieves information from the element definition table (155), segment definition table (156), element variables table (158), real option value table (162), risk reduction activity table (165), market sentiment table (166), the external factor definition table (169), the derivatives table (175), xml summary table (177), financial forecasts table (179) and factor variables table (176) as required to generate the organization value matrix (FIG. 8). The matrix is stored in the report table (164) and a summary version of the data is added to the xml summary table (177). The software in this block also creates and displays a summary Value Map™ Report for the enterprise (FIG. 9) via the report display and selection window (706). After the user (20) indicates that his or her review of the summary report is complete, processing advances to a block 405.

The software in block 405 prompts the user (20) via the analysis definition window (709) to specify changes in the enterprise that should be analyzed. The user (20) is given the option of: re-defining the structure for analysis purposes, examining the impact of changes in segments of value, components of value, elements of value and/or external factors on organization value by enterprise and/or optimizing a subset of the organization such as a segment of value, a component of value or a frame. For example, the user (20) may wish to:

1. optimize performance without considering the impact of market sentiment on organization value - this analysis would be completed by temporarily re-defining the structure and completing a new analysis;
2. optimize performance after adding the matrix of value for another enterprise or organization that may be purchased - this analysis would be completed by temporarily re-defining the structure and completing a new analysis;
3. forecast the the likely impact of a project on organization value - this analysis would be completed by mapping the expected results of the project to enterprise

segments of value, components of value, elements of value and/or external factors and recalculating value; or

4. maximize revenue from all enterprises in the organization – this analysis would be completed by defining a new model, the impact on the organization could be determined by using the output from the optimization to complete an analysis similar to the one described in item 3.

The software in block 405 saves the analysis definitions the user (20) specifies in the analysis definition table (178) in the application database (50) before processing advances to a software block 406.

The software in block 406 checks the analysis definition table (178) in the application database (50) to determine if the user (20) has specified a structure change analysis. If the calculation is a structure change analysis, then processing returns to block 205 and the processing described previously is repeated. Alternatively, if the calculation is not a structure change analysis, then processing advances to a software block 408.

The software in block 408 retrieves information from the xml summary table (177) and the analysis definition table (178) as required to determine what type of analysis will be completed and define a model for analysis. As mentioned previously, there are two types of analysis that may be completed by the software in this block – analyzing the impact of forecast changes and optimizing a subset of the enterprise. Analyzing the impact of changes to future values of external factors, segments of value, components of value, value drivers and/or elements of value requires recalculating value for the affected portions of organization value and comparing the new totals for the organization to the value information stored in the xml summary table (177). The results of this comparison, are then stored in the analysis definition table (178) before processing advances to software block 410. Alternatively, if the analysis involves optimizing a subset of the enterprise then the software in block 408 defines and initializes a probabilistic simulation model for the subset of the enterprise that is being analyzed. One embodiment of the probabilistic simulation models are Markov Chain Monte Carlo models, however, other simulation models can be used with similar results. The model is defined using the information retrieved from the xml summary table (177) and the analysis definition table (178) and then iterated as required to ensure the convergence of the frequency distribution of the output variables. After the calculation has been completed, the

software in block 408 saves the resulting information in the analysis definition table (178) before processing advances to software block 410.

The software in block 410 displays the results of any analyses with the report display and selection window (706) to the user (20). The user (20) optionally selects reports for display and/or printing. The format of the reports is either graphical, numeric or both depending on the type of report the user (20) specified in the system settings table (140). If the user (20) selects any reports for printing, then the information regarding the selected reports is saved in the reports table (164). After the user (20) has finished selecting reports, the selected reports are displayed to the user (20). After the user (20) indicates that the review of the reports has been completed, processing advances to a software block 411.

The software in block 411 checks the reports tables (164) to determine if any reports have been designated for printing. If reports have been designated for printing, then processing advances to a block 415. It should be noted that in addition to standard reports like the matrix of value and Value Map™ reports the system of the present invention can generate reports that rank the elements, external factors and/or the risks in order of their importance to overall value. The system can also produce "metrics" reports by tracing the historical measures for value drivers over time. The software in block 415 sends the designated reports to the printer (118). After the reports have been sent to the printer (118), processing advances to a software block 417. Alternatively, if no reports were designated for printing, then processing advances directly from block 411 to block 417.

The software in block 417 checks the system settings table (140) to determine if the system is operating in a continuous run mode. If the system is operating in a continuous run mode, then processing returns to block 205 and the processing described previously is repeated in accordance with the frequency specified by the user (20) in the system settings table (140). Alternatively, if the system is not running in continuous mode, then the processing advances to a block 418 where the system stops.

Thus, the reader will see that the system and method described above transforms extracted transaction data, corporate information and information from the Internet into a matrix of value for an enterprise. The system and method described above goes on to

use the detailed valuation and risk analysis information to identify an optimal risk reduction strategy before going on to define the efficient frontier for corporate financial performance. The level of detail, breadth and speed of the integrated analysis of performance allows users of the system to manage their operations in an fashion that is superior to many existing solutions.

While the above description contains many specificities, these should not be construed as limitations on the scope of the invention, but rather as an exemplification of one embodiment thereof. Accordingly, the scope of the invention should be determined not by the embodiment illustrated, but by the appended claims and their legal equivalents.